Alibaba Cloud

Apsara Stack Enterprise

Cloud Defined Storage User Guide

Product Version: V3.18.1 Document Version: 20240703

C-J Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

- You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
- 2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
- 3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
- 4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
- 5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
- 6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
🕂 Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	Danger: Resetting will result in the loss of user configuration data.
O Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
C) Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	Notice: If the weight is set to 0, the server no longer receives new requests.
? Note	A note indicates supplemental instructions, best practices, tips, and other content.	? Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings> Network> Set network type.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click OK .
Courier font	Courier font is used for commands	Run the cd /d C:/window command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	bae log listinstanceid Instance_ID
[] or [a b]	This format is used for an optional value, where only one item can be selected.	ipconfig [-all -t]
{} or {a b}	This format is used for a required value, where only one item can be selected.	switch {active stand}

Table of Contents

1.CDS	28
2.0SS	29
2.1. What is OSS?	29
2.1.1. Features	30
2.1.1.1. Manage buckets	30
2.1.1.1.1. Create a bucket	30
2.1.1.1.2. ACL	30
2.1.1.1.3. Static website hosting	30
2.1.1.1.4. Logging	31
2.1.1.1.5. Lifecycle rules	31
2.1.1.1.6. Bucket inventory	32
2.1.1.2. Manage objects	34
2.1.1.2.1. Upload objects	34
2.1.1.2.2. ACL	34
2.1.1.2.3. Download objects	34
2.1.1.2.4. Search for objects	34
2.1.1.2.5. Manage objects by using directories	35
2.1.1.2.6. Object tagging	35
2.1.1.3. Data security	37
2.1.1.3.1. Erasure coding	37
2.1.1.3.2. Retention policies	37
2.1.1.3.3. Resource isolation	40
2.1.1.3.4. Disaster recovery	40
2.1.1.3.5. Access permissions and account authorization	40
2.1.1.3.6. Server-side encryption	40
2.1.1.3.7. Client-side encryption	41

2.1.1.3.8. Versioning	42
2.1.1.4. Data processing	43
2.1.1.4.1. IMG	43
2.1.1.4.2. Video snapshots	44
2.2. Usage notes	44
2.3. Quick start	44
2.3.1. Log on to the OSS console	45
2.3.2. Create a bucket	45
2.3.3. Upload objects	46
2.3.4. Obtain object URLs	47
2.4. Buckets	47
2.4.1. View bucket information	47
2.4.2. Delete buckets	47
2.4.3. Modify bucket ACLs	48
2.4.4. Configure static website hosting	48
2.4.5. Configure hotlink protection for a bucket	48
2.4.6. Configure CORS	49
2.4.7. Configure lifecycle rules	50
2.4.8. Configure storage quota	50
2.4.9. Configure bucket tagging	51
2.4.10. Configure zone-disaster recovery	51
2.4.11. Versioning	52
2.4.12. Configure server-side encryption	53
2.4.13. Bind a VPC	53
2.4.14. Configure CRR rules	54
2.4.15. Configure cross-cloud replication	54
2.4.16. Configure retention policies	55
2.4.17. Log management	56

2.4.17.1. Configure logging	56
2.4.17.2. Real-time log query	56
2.4.18. Image processing	57
2.4.18.1. Configure image styles	57
2.4.18.2. Configure source image protection	58
2.4.19. Grants permissions to a role	58
2.5. Objects	60
2.5.1. Search for objects	60
2.5.2. Configure object ACLs	61
2.5.3. Configure object metadata	61
2.5.4. Create directories	62
2.5.5. Download objects	63
2.5.6. Delete objects	63
2.5.7. Manage parts	63
2.5.8. Configure object tagging	63
2.5.9. Configure bucket policies to authorize other users to	64
2.6. Add OSS paths	66
2.7. Create single tunnels	66
2.8. CSG	67
2.8.1. What is CSG?	67
2.8.2. Usage notes	68
2.8.3. Limits	69
2.8.4. Quick start	69
2.8.5. File gateways	71
2.8.5.1. Manage file gateways	71
2.8.5.2. Manage shares	72
2.8.5.3. Manage cache disks	75
2.8.5.4. Create an SMB user	75

2.8.5.5. Access shares	75
2.8.5.5.1. Access an NFS share	75
2.8.5.5.2. Access an SMB share	76
2.8.5.6. Update a gateway	77
2.8.5.7. Appendix	77
2.8.5.7.1. Permissions required by a gateway to operat	77
2.8.6. Configure an alert rule to monitor CSG gateways in	78
3.EBS	80
3.1. EBS	80
3.1.1. Features	81
3.1.1.1. Overview	81
3.1.1.2. Triplicate storage	81
3.1.1.3. Erasure coding	82
3.1.1.4. EBS device encryption	82
3.1.1.5. EBS device resizing	83
3.1.1.6. Snapshots	83
3.1.1.6.1. Overview	83
3.1.1.6.2. Mechanisms	83
3.1.1.6.3. Specifications of ECS Snapshot 2.0	84
3.1.1.7. Async replication	84
3.2. Log on to the Apsara Uni-manager Management Console	86
3.3. Getting started	87
3.3.1. Create a disk	87
3.3.2. Attach disks	88
3.3.3. Partition and format a disk	89
3.3.3.1. Format a data disk on a Linux instance	89
3.3.3.2. Format a data disk on a Windows ECS instance	91
3.4. Disks	91

3.4.1. Search for and view a disk9	91
3.4.2. Modify the properties of a disk	92
3.4.3. Modify the name and description of a disk	92
3.4.4. Resize a disk 9	92
3.4.5. Enable the multi-attach feature for disks that suppo	93
3.4.5.1. Overview of disks that support NVMe	93
3.4.5.2. Enable the multi-attach feature for disks	94
3.4.6. Encrypt a disk 9	96
3.4.6.1. Encrypt a system disk	9 6
3.4.6.2. Encrypt a data disk g	€9
3.4.7. Roll back a disk by using a snapshot	€9
3.4.8. Re-initialize disks	€9
3.4.8.1. Re-initialize a system disk) 7
3.4.8.2. Re-initialize a data disk	98
3.4.9. Change the category of a disk	99
3.4.10. Detach a data disk 10	00
3.4.11. Release a data disk 10)2
3.5. Snapshots 10)2
3.5.1. Disk snapshots 10)2
3.5.1.1. Create a snapshot 10)2
3.5.1.2. Search for and view a snapshot)3
3.5.1.3. Roll back a disk by using a snapshot 10)3
3.5.1.4. Create a custom image from a snapshot)4
3.5.1.5. Delete one or more snapshots at a time 10)4
3.5.2. Snapshot-consistent groups 10)4
3.5.2.1. Create a snapshot-consistent group)4
3.5.2.2. Search for and view a snapshot-consistent group)5
3.5.2.3. Roll back disks by using a snapshot-consistent g 10)6

3.5.2.4. Delete a snapshot-consistent group	106
3.6. Automatic snapshot policies	106
3.6.1. Create an automatic snapshot policy	106
3.6.2. Search for and view an automatic snapshot policy	107
3.6.3. Modify an automatic snapshot policy	108
3.6.4. Enable or disable an automatic snapshot policy	108
3.6.5. Delete an automatic snapshot policy	108
3.7. Async replication	108
3.7.1. Create a replication pair	108
3.7.2. View replication pairs	109
3.7.3. Modify a replication pair	109
3.7.4. Activate a replication pair to enable async replicatio	110
3.7.5. Stop a replication pair to disable async replication	110
3.7.6. Use the asynchronous replication feature to implem	111
3.7.7. Delete a replication pair	112
3.8. Storage sets	112
3.8.1. Create a storage set	112
3.8.2. View storage sets	113
3.8.3. Modify a storage set	113
3.8.4. Delete a storage set	113
3.9. Replication pair-consistent groups	113
3.9.1. Create a replication pair-consistent group	113
3.9.2. View replication pair-consistent groups	114
3.9.3. Modify a replication pair-consistent group	114
3.9.4. Add replication pairs to a replication pair-consistent	115
3.9.5. Remove replication pairs from a replication pair-cons	115
3.9.6. Activate a replication pair-consistent group to enable	115
	19-20-01

3.9.8. Use replication pair-consistent groups to implement	116
3.9.9. Delete a replication pair-consistent group	117
3.10. Appendix	117
3.10.1. Manage tags	117
4.Log Service	119
4.1. What is Log Service?	119
4.2. Quick start	119
4.2.1. Procedure	119
4.2.2. Log on to the Log Service console	120
4.2.3. Obtain an AccessKey pair	120
4.2.4. Manage a project	120
4.2.5. Manage a Metricstore	121
4.2.6. Manage Logstores	122
4.2.7. Manage shards	124
4.2.8. Terms	125
4.2.8.1. Terms	125
4.2.8.2. Log	127
4.2.8.3. Log group	127
4.2.8.4. Project	127
4.2.8.5. Logstore	128
4.2.8.6. Metricstore	128
4.2.8.7. Metric	128
4.2.8.8. Shard	128
4.2.8.9. Topic	129
4.2.9. Limits	130
4.2.9.1. Basic resources	130
4.2.9.2. Data read and write	131
4.2.9.3. Logtail	131

4.2.9.4. Data import	136
4.2.9.4.1. Limits on data import from OSS to Simple Lo	136
4.2.9.5. Data transformation	136
4.2.9.6. Query and analysis	138
4.2.9.7. Scheduled SQL	139
4.2.9.8. Alerting	141
4.3. Data collection	141
4.3.1. Collection by Logtail	141
4.3.1.1. Overview	141
4.3.1.1.1. Logtail overview	141
4.3.1.1.2. Log collection process of Logtail	143
4.3.1.1.3. Logtail configuration files and record files	144
4.3.1.2. Installation	148
4.3.1.2.1. Install Logtail on a Linux server	148
4.3.1.2.2. Install Logtail on a Windows server	150
4.3.1.2.3. Install Logtail components in a Kubernetes cl	151
4.3.1.2.4. Upgrade Logtail components in a Kubernetes	155
4.3.1.2.5. Configure the startup parameters of Logtail	156
4.3.1.3. Logtail machine group	163
4.3.1.3.1. Overview	163
4.3.1.3.2. Create an IP address-based machine group	163
4.3.1.3.3. Create a custom identifier-based machine gro	165
4.3.1.3.4. View server groups	166
4.3.1.3.5. Modify a server group	167
4.3.1.3.6. View the status of a server group	167
4.3.1.3.7. Delete a machine group	167
4.3.1.3.8. Manage a Logtail configuration	168
4.3.1.4. Collect text logs	168

	4.3.1.4.1. Configure text log collection	168
	4.3.1.4.2. Collect logs in simple mode	171
	4.3.1.4.3. Collect logs in full regex mode	173
	4.3.1.4.4. Collect logs in delimiter mode	176
	4.3.1.4.5. Collect logs in JSON mode	180
	4.3.1.4.6. Collect logs in NGINX mode	182
	4.3.1.4.7. Collect logs in IIS mode	186
	4.3.1.4.8. Collect logs in Apache mode	190
	4.3.1.4.9. Configure parsing scripts	194
	4.3.1.4.10. Time formats	195
	4.3.1.4.11. Import historical log files	197
	4.3.1.4.12. Log topics	198
4	.3.1.5. Collect container logs	199
	4.3.1.5.1. Overview	199
	4.3.1.5.2. Install the Logtail component	200
	4.3.1.5.3. Use the Log Service console to collect contai	203
	4.3.1.5.4. Use the Log Service console to collect contai	210
	4.3.1.5.5. Use CRDs to collect container logs in Daemo	220
	4.3.1.5.6. Use CRDs to collect container text logs in Si	226
	4.3.1.5.7. Use the Log Service console to collect contai	234
	4.3.1.5.8. Collect logs from standard Docker containers	237
	4.3.1.5.9. Collect Kubernetes events	239
	4.3.1.5.10. Collect container text logs	241
	4.3.1.5.11. Collect container stdout and stderr logs	245
	4.3.1.5.12. Collect standard Docker logs	251
4	.3.1.6. Custom plug-ins	254
	4.3.1.6.1. Collect MySQL binary logs	254
	4.3.1.6.2. Collect MySQL query results	260

4.3.1.6.3. Collect syslogs	264
4.3.1.6.4. Customize Logtail plug-ins to process data	267
4.3.1.7. Limits	282
4.3.2. Other collection methods	284
4.3.2.1. Use the web tracking feature to collect logs	284
4.3.2.2. Use SDKs to collect logs	286
4.3.2.2.1. Producer Library	286
4.3.2.2.2. Log4j Appender	286
4.3.2.2.3. Logback Appender	286
4.3.2.2.4. Golang Producer Library	287
4.3.2.2.5. Python logging	287
4.3.2.3. Collect common logs	289
4.3.2.3.1. Collect Log4j logs	289
4.3.2.3.2. Collect Python logs	290
4.3.2.3.3. Collect Node.js logs	293
4.3.2.3.4. Collect WordPress logs	293
4.3.2.3.5. Collect Unity3D logs	294
4.3.2.4. Data import	295
4.3.2.4.1. Import data from OSS to Simple Log Service	295
4.3.2.4.2. Time formats	298
4.4. Query and analysis	299
4.4.1. Log search overview	299
4.4.2. Log analysis overview	300
4.4.3. Reserved fields	302
4.4.4. Configure indexes	303
4.4.5. Query and analyze logs	305
4.4.6. Download logs	307
4.4.7. Enable Dedicated SQL	307

4.4.8. Index data type	308
4.4.8.1. Overview	308
4.4.8.2. Text type	308
4.4.8.3. Numeric type	309
4.4.8.4. JSON type	310
4.4.9. Query syntax and functions	312
4.4.9.1. Search syntax	312
4.4.9.2. LiveTail	316
4.4.9.3. LogReduce	317
4.4.9.4. Contextual query	319
4.4.9.5. Saved search	320
4.4.9.6. Quick analysis	321
4.4.10. SQL syntax and functions	323
4.4.10.1. General aggregate functions	323
4.4.10.2. Security check functions	324
4.4.10.3. Map functions and operators	326
4.4.10.4. Approximate functions	332
4.4.10.5. Mathematical statistics functions	333
4.4.10.6. Mathematical calculation functions	334
4.4.10.7. String functions	335
4.4.10.8. Date and time functions	337
4.4.10.9. URL functions	340
4.4.10.10. Regular expression functions	341
4.4.10.11. JSON functions	341
4.4.10.12. Type conversion functions	343
4.4.10.13. IP functions	344
4.4.10.14. GROUP BY clause	344
4.4.10.15. Window functions	345

4.4.10.16. HAVING clause	346
4.4.10.17. ORDER BY clause	347
4.4.10.18. LIMIT syntax	347
4.4.10.19. Conditional expressions	347
4.4.10.20. Nested subquery	349
4.4.10.21. Array functions and operators	349
4.4.10.22. Binary string functions	361
4.4.10.23. Bitwise functions	361
4.4.10.24. Interval-valued comparison and periodicity-valu	362
4.4.10.25. Comparison functions and operators	364
4.4.10.26. Lambda expressions	365
4.4.10.27. Logical functions	366
4.4.10.28. Column aliases	367
4.4.10.29. JOIN queries on a Logstore and a MySQL data	367
4.4.10.30. Geospatial functions	368
4.4.10.31. Geography functions	369
4.4.10.32. JOIN clause	370
4.4.10.33. UNNEST clause	370
4.4.11. Machine learning syntax and functions	371
4.4.11.1. Overview	371
4.4.11.2. Smooth functions	372
4.4.11.3. Multi-period estimation functions	375
4.4.11.4. Change point detection functions	376
4.4.11.5. Maximum value detection function	378
4.4.11.6. Prediction and anomaly detection functions	378
4.4.11.7. Time series decomposition function	382
4.4.11.8. Time series clustering functions	382
4.4.11.9. Frequent pattern statistics function	385

4.4.11.10. Differential pattern statistics function	385
4.4.11.11. Root cause analysis function	386
4.4.11.12. Correlation analysis functions	388
4.4.11.13. Kernel density estimation function	389
4.4.12. Scheduled SQL	390
4.4.12.1. How Scheduled SQL works	390
4.4.12.2. Limits	391
4.4.12.3. Create a Scheduled SQL job	393
4.4.12.3.1. Process and store data from a Logstore to	393
4.4.12.4. Manage a Scheduled SQL job	395
4.4.12.5. Query the result data of a Scheduled SQL job	396
4.4.12.6. Create a RAM role and grant the required perm	397
4.4.12.7. Syntax of time expressions	397
4.4.12.8. Time zones	398
4.4.12.9. FAQ	400
4.4.13. Advanced analysis	400
4.4.13.1. Optimize queries	400
4.4.13.2. Use cases	401
4.4.13.3. Examples of time field conversion	403
4.4.14. Associate Log Service with external data sources	403
4.4.14.1. Overview	403
4.4.14.2. Associate Log Service with a MySQL database	404
4.4.14.3. Associate Log Service with an OSS bucket	406
4.4.14.4. Associate Log Service with a hosted CSV file	406
4.4.15. Visual analysis	408
4.4.15.1. Charts	408
4.4.15.1.1. Chart overview	408
4.4.15.1.2. Display query results in a table	409

4.4.15.1.3. Display query results on a line chart 4	10
4.4.15.1.4. Display query results on a column chart 4	11
4.4.15.1.5. Display query results on a bar chart 4	12
4.4.15.1.6. Display query results on a pie chart 4	12
4.4.15.1.7. Display query results on an area chart 4	13
4.4.15.1.8. Display query results on a single value chart 4	14
4.4.15.1.9. Display query results on a progress bar 4	16
4.4.15.1.10. Display query results on a map	17
4.4.15.1.11. Display query results on a flow chart 4	18
4.4.15.1.12. Display query results in a Sankey diagram 4	18
4.4.15.1.13. Display query results on a word cloud 4	19
4.4.15.1.14. Display query results on a treemap chart 4	120
4.4.15.2. Charts (Pro) 4	120
4.4.15.2.1. Overview of charts (Pro) 4	120
4.4.15.2.2. Add a chart (Pro) to a dashboard 4	121
4.4.15.2.3. Attributes of charts (Pro) 4	121
4.4.15.2.4. Drill-down events 4	122
4.4.15.2.5. Variables 4	125
4.4.15.2.6. Table (Pro) 4	126
4.4.15.2.7. Line chart (Pro) 4	128
4.4.15.2.8. Flow chart (Pro) 4	131
4.4.15.2.9. Column chart (Pro) 4	133
4.4.15.2.10. Single value chart (Pro) 4	135
4.4.15.2.11. Pie chart (Pro) 4	138
4.4.15.3. Dashboard 4	139
4.4.15.3.1. Overview 4	139
4.4.15.3.2. Create and delete a dashboard4	139
4.4.15.3.3. Manage a dashboard in display mode4	140

4.4.15.3.4. Manage a dashboard in edit mode	441
4.4.15.3.5. Configure a drill-down event	442
4.4.15.3.6. Add a filter	446
4.4.15.3.7. Manage a Markdown chart	447
4.5. Data transformation	449
4.5.1. Data transformation overview	449
4.5.2. Terms	450
4.5.3. Data transformation basics	452
4.5.4. Limits	454
4.5.5. Configure preview modes	456
4.5.6. Create a data transformation job	456
4.5.7. Manage a data transformation job	458
4.5.8. Data transformation syntax	459
4.5.8.1. Language overview	459
4.5.8.2. Syntax overview	460
4.5.8.3. Data structures	462
4.5.8.4. Basic syntax	465
4.5.8.5. Function overview	467
4.5.8.6. Global processing functions	473
4.5.8.6.1. Overview of global processing functions	473
4.5.8.6.2. Flow control functions	474
4.5.8.6.3. Event processing functions	477
4.5.8.6.4. Field processing functions	483
4.5.8.6.5. Value extraction functions	487
4.5.8.6.6. Mapping and enrichment functions	500
4.5.8.6.7. Value-added content function	507
4.5.8.7. Expression functions	510
4.5.8.7.1. Overview of expression functions	510

4.5.8.7.2. Event check functions	510
4.5.8.7.3. Operator functions	514
4.5.8.7.4. Conversion functions	528
4.5.8.7.5. Arithmetic functions	532
4.5.8.7.6. String functions	543
4.5.8.7.7. Date and time functions	568
4.5.8.7.8. Regular expression functions	582
4.5.8.7.9. Grok function	585
4.5.8.7.10. Structured data functions	588
4.5.8.7.11. IP address parsing functions	590
4.5.8.7.12. Encoding and decoding functions	600
4.5.8.7.13. List functions	614
4.5.8.7.14. Dictionary functions	616
4.5.8.7.15. Table functions	619
4.5.8.7.16. Resource functions	621
4.5.8.7.17. Parsing functions	633
4.5.8.8. General reference	635
4.5.8.8.1. Standard encoding formats	635
4.5.8.8.2. Query string syntax	637
4.5.8.8.3. Field extraction modes	640
4.5.8.8.4. Regular expressions	642
4.5.8.8.5. Grok patterns	643
4.5.8.8.6. JMESPath syntax	654
4.5.8.8.7. Date and time formatting directives	656
4.5.8.8.8. Time zones	658
4.6. Alerts	660
4.6.1. Overview	660
4.6.2. Configure an alarm	661

	4.6.2.1. Configure an alert rule	661
	4.6.2.2. Authorize a RAM user to manage alert rules	662
	4.6.2.3. Configure alert notification methods	663
4	4.6.3. Modify and view an alarm	665
	4.6.3.1. Modify an alert rule	665
	4.6.3.2. View alert statistics	665
	4.6.3.3. Manage alerts	666
4	4.6.4. Relevant syntax and fields for reference	667
	4.6.4.1. Syntax of conditional expressions in alert rules	667
	4.6.4.2. Fields in alert logs	668
4	4.6.5. FAQ	670
	4.6.5.1. A DingTalk alert notification fails to be sent and	670
4.7	7. Real-time consumption	671
4	4.7.1. Overview	671
4	4.7.2. Consume log data	671
4	4.7.3. Consumption by consumer groups	672
	4.7.3.1. Use consumer groups to consume log data	672
	4.7.3.2. View the status of a consumer group	676
4	4.7.4. Use Storm to consume log data	677
4	4.7.5. Use Flume to consume log data	680
4	4.7.6. Use open source Flink to consume log data	682
4	4.7.7. Use Logstash to consume log data	685
4	4.7.8. Use Spark Streaming to consume log data	686
4	4.7.9. Use Realtime Compute to consume log data	689
4.8	3. Data shipping	691
4	4.8.1. Ship logs to OSS	691
	4.8.1.1. Overview	691
	4.8.1.2. Ship log data from Log Service to OSS	691

4.8.1.3. Obtain the ARN of a RAM role	693
4.8.1.4. Storage Formats	694
4.8.1.5. Decompress Snappy compressed files	695
4.9. Log applications	696
4.9.1. Trace	696
4.9.1.1. Usage notes	696
4.9.1.2. Trace data formats	697
4.9.1.3. Create a trace instance	698
4.9.1.4. Import trace data	698
4.9.1.4.1. Overview	698
4.9.1.4.2. New import methods	700
4.9.1.4.2.1. Import trace data from Java applications t	700
4.9.1.4.2.2. Import trace data from Golang application	702
4.9.1.4.2.3. Import trace data from Python application	708
4.9.1.4.2.4. Import trace data from Node.js application	713
4.9.1.4.2.5. Import trace data from C# applications to	717
4.9.1.4.2.6. Import trace data from Rust applications	719
4.9.1.4.2.7. Import trace data from Ruby applications	721
4.9.1.4.2.8. Import trace data from PHP applications t	722
4.9.1.4.2.9. Import trace data from C++ applications	723
4.9.1.4.2.10. Import trace data from Android apps to	725
4.9.1.4.2.11. Import trace data from iOS apps to Log	732
4.9.1.4.3. Existing import methods	735
4.9.1.4.3.1. Import trace data from OpenCensus to Lo	735
4.9.1.4.3.2. Import trace data from Zipkin to Log Ser	736
4.9.1.4.3.3. Import trace data from Apache SkyWalkin	737
4.9.1.4.3.4. Import trace data from OpenTelemetry to	739
4.9.1.4.3.5. Import trace data from Jaeger to Log Ser	740

4.9.1.4.4. View the import results of trace data	742
4.9.1.5. View the details of a trace instance	743
4.9.1.6. Query and analyze trace data	743
4.9.1.7. View trace details	744
4.9.1.8. Best practices	745
4.9.1.8.1. Import trace data from Log Service to Grafan	745
4.9.1.8.2. Import trace data from Apache SkyWalking to	747
4.9.1.8.3. Import Ingress trace data from Kubernetes cl	748
4.9.1.9. FAQ	750
4.9.1.9.1. How do I implement OpenTelemetry automati	750
4.10. Time series storage	752
4.10.1. Data import	752
4.10.1.1. Collect metric data from hosts	752
4.10.1.2. Collect ping and tcping data	755
4.10.1.3. Import metrics collected by Telegraf	758
4.10.1.3.1. Telegraf overview	758
4.10.1.3.2. Collect metric data from Elasticsearch cluste	758
4.10.1.3.3. Collect metric data from MySQL servers	759
4.10.1.3.4. Collect metric data from Redis databases	760
4.10.1.3.5. Collect metric data from MongoDB databases	761
4.10.1.3.6. Collect metric data from ClickHouse databas	762
4.10.1.3.7. Collect metric data from Kafka servers	763
4.10.1.3.8. Collect metric data from Java applications o	764
4.10.1.3.9. Collect metric data from NGINX servers	766
4.10.1.3.10. Collect metric data from NVIDIA GPUs	767
4.10.1.4. Collect metric data from Prometheus	768
4.10.1.4.1. Collect metric data from Prometheus by usi	768
4.10.1.4.2. Collect metric data from Prometheus by usi	770

4.10.2. Query and analysis77	71
4.10.2.1. Overview of query and analysis of time series	71
4.10.2.2. Query and analyze time series data 77	72
4.10.3. Visualization 77	73
4.10.3.1. Configure a time series chart	73
4.10.3.2. Send time series data from Log Service to Gra 77	74
4.11. RAM 77	75
4.11.1. Permissions required by a RAM user to manage Lo	75
4.11.2. Use custom policies to grant permissions to a RAM	75
4.12. Monitor Log Service77	78
4.12.1. Overview 77	78
4.12.2. Manage service logs	79
4.12.3. Log types 78	80
4.12.4. Service log dashboards78	85
4.13. FAQ 78	85
4.13.1. Log collection 78	85
4.13.1.1. How do I troubleshoot errors that occur when I 78	85
4.13.1.2. What can I do if Log Service does not receive	86
4.13.1.3. How do I query the status of local log collectio 78	87
4.13.1.4. How do I debug a regular expression? 79	95
4.13.1.5. How do I optimize regular expressions?	96
4.13.1.6. How do I use the full regex mode to collect lo 79	96
4.13.1.7. How do I specify time formats for logs?	96
4.13.1.8. How do I configure non-printable characters in	97
4.13.1.9. How do I troubleshoot errors that occur when I	97
4.13.1.10. How do I obtain the labels and environment v 79	99
4.13.2. Log search and analysis80	00
4.13.2.1. FAQ about log query80	00

4.13.2.2. What can I do if I cannot obtain the required r	801
4.13.2.3. What are the differences between log consump	802
4.13.2.4. How do I resolve common errors that occur wh	802
4.13.2.5. Why data queries are inaccurate?	803
4.13.2.6. How do I configure indexes for historical log da	803
4.13.3. Alarm	804
4.13.3.1. FAQ about alerts	804
4.13.4. What do I do if the Forbidden.SLS::ListProject error	805
5.Tablestore	806
5.1. Terms	806
5.2. Limits	806
5.3. Quick start	807
5.3.1. Log on to the Tablestore console	807
5.3.2. Create a Tablestore instance	808
5.3.3. Create a data table	809
5.3.4. Read and write data in the console	811
5.3.5. Bind a VPC to a Tablestore instance	813
5.3.6. Use Tunnel Service	813
5.4. Secondary index	815
5.4.1. Usage notes	815
5.4.2. Use secondary indexes	817
5.4.3. Common scenarios	819
5.5. Search index	826
5.5.1. Overview	826
5.5.2. Features	828
5.5.3. Limits	829
5.5.4. Data type mappings	831
5.5.5. Basic features	832

5.5.5.1. Search indexes	832
5.5.5.2. Lifecycle management	834
5.5.5.3. Types of date data	835
5.5.5.4. ARRAY and NESTED field types	837
5.5.5.5. Tokenization	838
5.5.6. Advanced features	840
5.5.6.1. Virtual columns	840
5.5.6.2. Dynamically modify schemas	842
5.5.6.3. Fuzzy query	843
6.Hybrid Disaster Recovery	846
6.1. What is Hybrid Disaster Recovery?	846
6.2. Getting started	846
6.2.1. Procedure	846
6.2.2. Log on to the HDR console	846
6.2.3. Terms	847
6.3. Disaster recovery in the cloud	847
6.3.1. Cross-zone disaster recovery	847
6.3.1.1. Process overview	848
6.3.1.2. Async replication	850
6.3.1.2.1. Step 1: Plan resources	850
6.3.1.2.2. Step 2: Create a site pair	850
6.3.1.2.3. Step 3: Configure network and security settin	851
6.3.1.2.4. Step 4: Create a protection group	852
6.3.1.2.5. Step 5: Add instances to be protected	853
6.3.1.2.6. Step 6: Start replication	854
6.3.1.2.7. Step 7: Perform a failover	854
6.3.1.3. CDR	856
6.3.1.3.1. Limits	856

6.3.1.3.2. Step 1: Plan resources	857
6.3.1.3.3. Step 2: Create a site pair	857
6.3.1.3.4. Step 3: Configure network and security settin	858
6.3.1.3.5. Step 4: Add instances to be protected	859
6.3.1.3.6. Step 5: Start replication	860
6.3.1.3.7. Step 6: Perform a failover	860
6.3.2. Cross-cloud disaster recovery	863
6.3.2.1. Cross-cloud disaster recovery	863
6.3.2.2. Step 1: Plan resources	863
6.3.2.3. Step 2: Create a site pair	864
6.3.2.4. Step 3: Configure network and security settings	865
6.3.2.5. Step 4: Create a protection group	865
6.3.2.6. Step 5: Add instances to be protected	866
6.3.2.7. Step 6: Start replication	867
6.3.2.8. Step 7: Perform a failover	867
6.4. Health center	869
6.4.1. View current alerts	869
6.4.2. View the alert history	869
6.4.3. Manage inspection tasks	869
6.4.4. Manage O&M tickets	870
6.5. Cloud configuration	870
7.CDS CM	872
7.1. CDS Configuration Manager	872
7.2. Log on to the Apsara Uni-manager Management Console	872
7.3. Basic settings	873
7.3.1. Configure organizations	873
7.3.2. Initialize CDS CM	874
7.4. Product overview	874

7.5. Resource management	875
7.5.1. Manage tenant resources	875
7.5.2. View resource topology	876
7.6. Monitoring reports	877
7.6.1. Manage resource reports	877
7.7. Intelligent analysis	877
7.7.1. View security assessment information	877
7.8. System management	878
7.8.1. Manage tenants	878
7.8.2. Manage tasks	878

1.CDS

Cloud Defined Storage (CDS) is a distributed file system that provides storage based on cloud services. CDS is secure, cost-efficient, and highly reliable. You can centrally manage resources in CDS and flexibly scale resources on and off the cloud. This facilitates local data storage and retrieval.

Based on Object Storage Service (OSS), Elastic Block Storage (EBS), and Simple Log Service, CDS provides storage for unstructured data such as files, images, and videos, and supports log storage, query, and analysis features. This allows users in different industries to access unstructured data and process massive amounts of logs. CDS is ideal to provide unstructured data storage and log services in big data scenarios such as mobile application and large websites. CDS offers a one-stop storage solution for users in different industries that is cost-effective, secure, and reliable.

2.OSS 2.1. What is OSS?

Object Storage Service (OSS) is a secure, cost-effective, and highly reliable cloud storage service provided by Alibaba Cloud.

Overview

OSS is an object storage service based on key-value pairs. Files uploaded to OSS are stored as objects in buckets. You can obtain the content of an object based on the object key.

Compared with user-created server storage, OSS has outstanding advantages in reliability, security, cost-effectiveness, and data processing capabilities. OSS enables you to store and retrieve a variety of unstructured data objects, such as text, images, audios, and videos over networks anytime.

In OSS, you can perform the following operations:

- Create a bucket and upload objects to the bucket.
- · Obtain an object URL from OSS to share or download the object.
- Modify the attributes or metadata of a bucket or an object. You can also configure the access control list (ACL) of the bucket or the object.
- Perform basic and advanced operations in the OSS console.
- Perform basic and advanced operations by using OSS SDKs or calling RESTful API operations in your application.

Basic concepts

Object

The basic unit for data operations in OSS. Objects are also known as OSS files. An object is composed of object metadata, object content, and a key. A key can uniquely identify an object in a bucket. Object metadata is a group of key-value pairs that define the properties of an object, such as the last modification time and the object size. You can also assign user metadata to the object.

The lifecycle of an object starts when the object is uploaded and ends when it is deleted. During the lifecycle, the object cannot be modified. OSS does not support modifying objects. If you want to modify an object, you must upload a new object with the same name as the existing object to replace it.

(?) Note Unless otherwise stated, objects and files mentioned in OSS documents are collectively called objects.

Bucket

A container for OSS objects. Each object in OSS is contained in a bucket. You can configure and modify the attributes of a bucket to manage ACLs and lifecycle rules of the bucket. These attributes apply to all objects in the bucket. Therefore, you can create different buckets to meet different management requirements.

- OSS does not use a hierarchical structure for objects, but instead uses a flat structure. All elements are stored as objects in buckets. However, OSS
 supports folders as a concept to group objects and simplify management.
- You can create multiple buckets.
- A bucket name must be unique in OSS within an Apsara Stack tenant account. Bucket names cannot be changed after the buckets are created.
 A bucket can contain an unlimited number of objects.
- Strong consistency

Object operations in OSS are atomic. Operations can either succeed or fail without intermediate states. To ensure that users can access only complete data, OSS does not return corrupted or partial data.

Object-related operations in OSS are highly consistent. For example, when a user receives an upload (PUT) success response, the uploaded object can be read immediately, and copies of the object are written to multiple devices for redundancy. Therefore, there are no situations where data is not obtained when you perform the read-after-write operation. The same is true for delete operations. After a user deletes an object, the object and its copies no longer exist.

Similar to traditional storage devices, modifications are immediately visible in OSS and consistency is guaranteed.

Comparison between OSS and file systems

OSS is a distributed object storage service that stores objects based on key-value pairs. You can retrieve object content based on unique object keys. For example, the object name test1/test.jpg does not necessarily indicate that the object is stored in a directory named test1. In OSS, test1/test.jpg is only a string. There is nothing essentially different between test1/test.jpg and a.jpg. Therefore, similar amounts of resources are consumed regardless of which object you access.

A file system uses a typical tree index structure. To access a file named test1/test.jpg, you must first access the test1 directory and then search for the test.jpg file in this directory. This makes it easy for a file system to support folder operations, such as renaming, deleting, and moving directories because these operations are only performed on directories. However, the performance of a file system depends on the capacity of a single device. The more files and directories that are created in the file system, the more resources and time are consumed.

You can simulate similar folder functions of a file system in OSS, but such operations are costly. For example, if you want to rename the test1 directory as test2, OSS must copy all objects whose names start with test1/ to generate objects whose names start with test2. This operation consumes a large amount of resources. Therefore, we recommend that you do not perform such operations in OSS.

Objects stored in OSS cannot be modified. A specific operation must be called to append an object. The generated objects are different from the objects uploaded by using other methods. To modify an object, you must upload the entire object again. A file system allows you to modify files. You can modify the content at a specified offset location or truncate the end of a file. These features make file systems suitable for more general scenarios. However, OSS supports a large amount of concurrent access, whereas the performance of a file system is subject to the performance of a single device.

We recommend that you do not map operations on OSS objects to file systems because it is inefficient. If you attach OSS as a file system, we recommend that you only add new files, delete files, and read files. You can make full use of OSS advantages, such as the capability to process and store large amounts of unstructured data such as images, videos, and documents.

Limits

Limit	Description
Bucket	You can create a maximum of 100 buckets.After a bucket is created, its name and region cannot be modified.

Upload objects	 Objects larger than 5 GB cannot be uploaded by using the following methods: console upload, simple upload, form upload, or append upload. To upload an object that is larger than 5 GB, you must use multipart upload. The size of an object uploaded by using multipart upload cannot exceed 48.8 TB. If you upload an object that has the same name as an existing object in OSS, the new object overwrites the existing object.
Delete objects	 Deleted objects cannot be recovered. You can delete up to 100 objects at a time from the console. To delete more than 100 objects at a time, you must call API operations or use an OSS SDK.
Lifecycle rules	You can configure up to 1,000 lifecycle rules for each bucket.

2.1.1. Features

2.1.1.1. Manage buckets

2.1.1.1.1. Create a bucket

A bucket is a container that is used to store objects in Object Storage Service (OSS). Every object is contained in a bucket. You can configure a variety of bucket attributes such as the region, access control list (ACL), and storage class. You can create buckets of different storage classes to store data.

Naming conventions

After a bucket is created, the name of the bucket cannot be modified. OSS supports the following bucket naming conventions:

- The name of a bucket must be unique in OSS in an Apsara Stack tenant account.
- The name can contain only lowercase letters, digits, and hyphens (-).
- The name must start and end with a lowercase letter or a digit.
- The name must be 3 to 63 characters in length.

Examples

The following examples of bucket names are valid:

- examplebucket1
- test-bucket-2021
- aliyun-oss-bucket
- The following examples show invalid bucket names and the reasons why the names are invalid:
- Examplebucket1 (Uppercase letters are included.)
- test_bucket_2021 (Underscores (_) are included.)
- aliyun-oss-bucket- (The name ends with a hyphen (-).)

2.1.1.1.2. ACL

You can configure the access control list (ACL) of a bucket when you create the bucket or modify the ACL of a created bucket. Only the owner of a bucket can configure or modify the ACL of the bucket.

You can set one of the following three ACLs for a bucket:

ACL	Description
public-read-write	Anyone, including anonymous users, can perform read and write operations on the objects in the bucket. Warning All Internet users can access objects in the bucket and write data to the bucket. This may result in unexpected access to the data in your bucket and out-of-control costs. If a user uploads prohibited data or information, your legitimate interests and rights may be infringed. Therefore, we recommend that you do not set your bucket ACL to public read/write except in special cases.
public-read	Only the bucket owner can perform write operations on the objects in the bucket. Other users, including anonymous users can perform only read operations on the objects in the bucket.
private	Only the bucket owner can perform read and write operations on the objects in the bucket. Other users have no access to the objects in the bucket.

2.1.1.1.3. Static website hosting

Static websites are websites in which all web pages consist only of static content, including scripts such as JavaScript code that is run on the client. You can use the static website hosting feature to host your static website on an Object Storage Service (OSS) bucket and use the endpoint of the bucket to access the website.

Usage notes

When you configure static website hosting, you must specify the default homepage and the default 404 page for the website.

• The default homepage appears when you use a browser to access the static website hosted on an OSS bucket. The default homepage functions in a similar manner to the index.html file of a website.

The object that you specify as the default homepage must be an object that is stored in the root directory of the bucket and allows anonymous access. The object must be in the HTML format.

• The default 404 page is the error page returned by OSS. When you use a browser to access the static website hosted on an OSS bucket and a 404 error occurs, OSS returns the default 404 page.

The object that you specify as the default 404 page must be an object that is stored in the root directory of the bucket and allows anonymous access. The object must be in one of the following formats: HTML, JPG, PNG, BMP, and WebP.

Configurations

After you host a static website on a bucket, you must upload an object whose name is the same as that of the default homepage, such as index.html, to the bucket. If the bucket contains a directory such as subdir/, you must also upload the object named index.html to subdir/. In addition, you must upload an object whose name is the same as that of the default 404 page, such as error.html, to the bucket. The following structure shows the objects and directories in the sample bucket:

- Bucket index.html error.html example.txt subdir/
- ____ index.html

In this example, the custom domain name example.com is mapped to the bucket, the default homepage of the static website hosted on the bucket is index.html, and the default 404 page of the website is error.html. When you access the static website by using the custom domain name, OSS returns different responses based on your configurations of Static Pages for the bucket that hosts the website.

- When you access https://example.com/ and https://example.com/subdir/ ,OSS returns https://example.com/index.html .
- When you access https://example.com/example.txt, the example.txt object is obtained.
- When you access https://example.com/object , OSS returns https://example.com/error.html if the object object does not exist.

2.1.1.1.4. Logging

When you access Object Storage Service (OSS), large numbers of access logs are generated. After you enable and configure logging for a bucket, OSS generates log objects every hour in accordance with a predefined naming convention and then stores the access logs as objects in a specified bucket. You can use Apsara Stack Log Service or build a Spark cluster to analyze the logs.

Naming conventions for log objects

The following naming conventions apply to log objects that are stored in OSS:

<TargetPrefix><SourceBucket>YYYY-mm-DD-HH-MM-SS-UniqueString

Field	Description
TargetPrefix	The prefix of the log object name.
SourceBucket	The name of the source bucket for which access logs are generated.
YYYY-mm-DD-HH-MM-SS	The time when the log object is created. The items of this field indicate the year, month, day, hour, minute, and second in sequence.
UniqueString	The string generated by OSS to uniquely identify the log object.

Usage notes

- The source bucket for which access logs are generated and the destination bucket in which the log objects are stored can be the same bucket or different buckets. However, the destination bucket must belong to the same account in the same region as the source bucket.
- OSS generates bucket access logs on an hourly basis. However, requests in the previous hour may be recorded in the logs generated for the subsequent hour.
- Before you disable logging, OSS keeps generating access logs. Delete log objects that you no longer need based on lifecycle rules to reduce storage costs.
- OSS adds more fields to access logs in the future. We recommend that developers consider potential compatibility issues when they develop log
 processing tools.

2.1.1.1.5. Lifecycle rules

You can configure lifecycle rules to regularly delete expired objects and parts to reduce storage costs.

Scenarios

You can configure a lifecycle rule to regularly delete objects that are no longer accessed or convert the storage class of non-hot data to Infrequent Access (IA), Archive, or Cold Archive. This improves data management efficiency and saves storage costs. You can manually delete up to 1,000 objects each time. If a bucket contains more than 1,000 objects and you want to delete all objects from the bucket, you must delete the objects multiple times. In this case, you can configure a lifecycle rule to delete all objects in the bucket the next day. This way, all objects in the bucket can be deleted the next day.

Usage notes

· Number of lifecycle rules

You can configure up to 1,000 lifecycle rules for each bucket.

Effective time

After you configure a lifecycle rule, OSS loads the rule within 24 hours. After the lifecycle rule is loaded, OSS runs the rule every day at 08:00:00 (UTC+8) and completes the operations that are triggered by the rule within 24 hours. The interval between the last modified time of an object and the time when the lifecycle rule is run must be longer than 24 hours. For example, if you configure a lifecycle rule for a bucket to delete objects one day after they are uploaded, objects that are uploaded on July 20, 2020 are deleted on a different date based on the specific time when the objects are uploaded.

Objects uploaded before 08:00:00 (UTC+8) are deleted from 08:00:00 (UTC+8) on July 21, 2020 to 08:00:00 (UTC+8) on July 22, 2020.
 Objects uploaded after 08:00:00 (UTC+8) are deleted from 08:00:00 (UTC+8) on July 22, 2020 to 08:00:00 (UTC+8) on July 23, 2020.

① Important When you update a lifecycle rule, tasks to perform on the day based on the lifecycle rule are suspended. We recommend that you do not frequently update lifecycle rules.

Elements of a lifecycle rule

A lifecycle rule consists of the following elements:

• Policy: the policy used to match objects and parts.

- Match by prefix: Objects and parts are matched by prefix. You can create multiple rules to match objects with different object name prefixes. Each prefix must be unique.
- Match by tag: Objects are matched by tag key and tag value. You can specify multiple tags in a single lifecycle rule. The lifecycle rule applies to all
 objects that have the specified tags. Lifecycle rules cannot match parts by tag.
- Match by prefix and tag: Objects are matched by specified prefixes and tags.
- Match by bucket: The rule matches all objects and parts stored in the bucket. After you configure a lifecycle rule for a bucket to match all objects and parts in the bucket, other lifecycle rules cannot be configured for the bucket.
- Object lifecycle policy: specifies the validity period or the expiration date of objects and the operation to perform on expired objects.
- Validity period: A validity period is specified for objects in buckets for which versioning is disabled and the current versions of objects in buckets for which versioning is enabled. In addition, the operation to perform on these objects after they expire is specified. Objects that match the lifecycle rule are retained for the specified validity period after the objects are last modified. The specified operation is performed on these objects after they expire.
- Expiration date: An expiration date is specified for objects in buckets for which versioning is disabled and the current versions of objects in buckets for which versioning is enabled. In addition, the operation to perform on these objects after they expire is specified. All objects that are last modified before this date expire, and the specified operation is performed on these objects.
- Validity period of the previous versions of objects: A validity period is specified for the previous versions of objects. In addition, the operation to
 perform on these previous versions is specified. Objects that match the lifecycle rule are retained for the specified validity period after the object
 versions become the non-curent versions. The specified operation is performed on these objects after they expire.
- Part lifecycle policy: the policy used to specify the validity period or expiration date for parts and the operation to perform on these expired parts.
 Validity period: A validity period is specified for parts. Parts that match the lifecycle rule are retained within the validity period and are deleted
 - after they expire.
 Expiration date: An expiration date is specified for parts. Parts that are last modified before this date expire and are deleted.

2.1.1.1.6. Bucket inventory

You can use the bucket inventory feature to export information about specified objects in a bucket, such as the number, sizes, storage classes, and encryption status of the objects. To list a large number of objects, we recommend that you use the bucket inventory feature instead of the GetBucket (ListObjects) operation.



Overview

After an inventory is configured for a bucket, OSS generates inventory lists at the specified time interval. The following structure shows the directories in which generated inventory lists are stored.

dest bucket

-745a29e3-bfaa-490d-9109-47086afcc8f2.csv.gz

Directory	Description
destination-prefix/	This directory is generated based on the prefix specified for inventory lists. If no prefix is specified for inventory lists, this directory is omitted.
src_bucket/	This directory is generated based on the name of the source bucket for which inventory lists are generated.
inventory_id/	This directory is generated based on the name of the inventory.
YYYY-MM-DDTHH-MMZ/	This directory indicates the start time when the bucket is scanned. The name of this directory is a timestamp in UTC. Example: 2020-05-17T16-00Z. The manifest.json object and manifest.checksum object are stored in this directory.
data/	Inventory lists that include the list of objects in the source bucket and the metadata of exported objects in the source bucket are stored in this directory. Inventory lists are CSV objects that are compressed by using Gzip. Important • When a large number of objects are stored in the source bucket, OSS automatically splits the inventory lists into multiple CSV objects for downloading and processing. The names of the CSV objects are generated in the following format in sequence: uuid.csv.gz, uuid-1.csv.gz, and uuid-2.csv.gz. You can obtain the list of the CSV objects from the manifest.json object. Then, you can extract the objects based on the preceding sequence to read the inventory lists. • The information about a single object is not split into multiple inventory lists.

After an inventory is configured for a bucket, the following objects are generated based on the inventory:

Manifest objects

Manifest objects include manifest.json and manifest.checksum.

• manifest.json: stores the metadata of inventory lists and related information.

{
 "creationTimestamp": "1642994594",
 "destinationBucket": "destbucket",
 "fileFormat": "CSV",
 "fileSchema": "Bucket, Key, VersionId, IsLatest, IsDeleteMarker, Size, StorageClass, LastModifiedDate, ETag, IsMultipartUploaded, Encry
ptionStatus",
 "files": [{
 "MDSchecksum": "F77449179760C3B13F1E76110F07****",
 "key": "destbucket/inventory0124/data/a1574226-b5e5-40ee-91df-356845777c04.csv.gz",
 "size": 2046]],
 "sourceBucket": "srcbucket",
 "version": "S2019-09-01"}

The following table describes the fields included in the manifest.json object.

Field	Description
creationTimestamp	The start time when the source bucket is scanned. The value of this field is a UNIX timestamp.
destinationBucket	The destination bucket in which the inventory lists are stored.
fileFormat	The format of the inventory lists.
fileSchema	The fields contained in each inventory list. The values in the CSV objects are sorted based on the sequence of fields in fileSchema. You can resolve and read the row values in the CSV objects based on the sequence of fields in fileSchema.
files	Information about the name, size, and MD5 hash of each inventory list.
sourceBucket	The source bucket for which the inventory lists are generated.
version	The version of the inventory list.

• manifest.checksum: stores the MD5 hash of the manifest.json object. Example: 8420A430CBD6B659A1C0DFC1C11A*****

Inventory lists

Inventory lists contain the exported object information and are stored in the data/ directory. The following figure shows a sample inventory list.

aliyun HICHIT	zh-hz 1119	CAEQI UDh	FALSE	FALSE	429 Standard	2021-02-08T16-01-1B39 BD13	FALSE	FALSE
aliyun	zh-hz 1119	CAEQI Dt2r	TRUE	TRUE	0 Standard	2021-04-21T09-54-0 DDI	FALSE	FALSE
aliyun	zh-hz 1119	CAEQI	FALSE	FALSE	32 Standard	2021-02-09T16-01-3E88 269	FALSE	FALSE
aliyun Hari	zh-hz 1119	CAEQI Dw2	TRUE	TRUE	0 Standard	2021-04-21T09-54-0 DD[2021-04-21T09-54-0 DD[FALSE	FALSE
aliyun	zh-hz 1119	CAEQI //Dc	FALSE	FALSE	429 Standard	2021-02-09T16-01-34DC ?92E	FALSE	FALSE
aliyun Hari	zh-hznouwzr 1119	CAEQI 1D8	TRUE	TRUE	0 Standard	2021-04-21T09-54-0 DDE	FALSE	FALSE

The following table describes the fields in the preceding figure from left to right.

Field	Description
Bucket	The name of the bucket for which the inventory is created.
Кеу	The name of the object in the bucket. The object name is URL-encoded. You must decode the object name to obtain the object name.
VersionId	The version ID of the object. This field exists only when versioning is enabled for the bucket and the inventory specifies that all versions of data are exported.
lsLatest	Indicates whether the version is the latest version. If the version is the latest version, the value of this field irrue. If the version is not the latest version, the value of this field is False. This field exists only when versioning is enabled for the bucket and the inventory specifies that all versions of data are exported.
IsDeleteMarker	Indicates whether the version is a delete marker. If the version is a delete marker, the value of this field is rue. If the version is not a delete marker, the value of this field is False. This field exists only when versioning is enabled for the bucket and the inventory specifies that all versions of data are exported.
Size	The size of the object. Unit: bytes.
StorageClass	The storage class of the object.
LastModifiedDate	The time when the object is last modified.
ETag	 The ETag of the object. An ETag is generated when an object is created. The ETag is used to identify the content of the object. For an object that is created by calling the PutObject operation, the ETag value of the object is the MD5 hash of the object content. If an object is created by using other methods, the ETag of the object is not the MD5 hash of the object content but a unique value calculated based on the object.
IsMultipartUploaded	Indicates whether the object is created by using multipart upload. If the object is created by using multipart upload, the value of this field is True. If the object is not created by using multipart upload, the value is false.
EncryptionStatus	Indicates whether the object is encrypted. If the object is encrypted, the value of this field is rue. If the object is not encrypted, the value is False.

Usage notes

Recommended configurations

- If less than 10 billion objects are stored in a bucket, we recommend that you export inventory lists on a daily or weekly basis based on your business requirements.
- If more than 10 billion objects are stored in a bucket, we recommend you export inventory lists on a weekly basis.

Traffic and bandwidth

To increase the speed at which inventory lists are exported, bucket-level and user-level bandwidth may be consumed when the inventory lists are exported to the destination bucket. If the bucket for which you want to configure the inventory is frequently accessed and the available bandwidth of the bucket is limited, we recommend that you create a destination bucket to store the inventory lists.

Exceptions

- If no objects are stored in the bucket for which the inventory is configured or no objects matches are found for the specified prefix in the inventory, inventory lists are not generated.
- When you export the inventory lists, the exported lists may not contain all matching objects in the source bucket due to operations such as creation, deletion, or overwriting. If the time when an object is last modified is earlier than the time specified by the createTimeStamp field in the manifest.json object, inventory lists contain the information about the object. Otherwise, inventory lists may not contain information about the object. We recommend that you check the object attributes by calling the HeadObject operation before you export information about an object.

Deletion of inventory lists

Before you delete an inventory, OSS continuously exports inventory lists on a daily or weekly basis based on the inventory. To prevent OSS from generating unnecessary inventory lists, you can delete inventories that you no longer need in a timely manner. You can also delete exported historical inventory lists that you no longer need.

Limits

- You can configure up to 1,000 inventory for a bucket by using OSS SDK.
- The bucket for which you want to configure the inventory can be different from the bucket in which you want to store the generated inventory lists. However, the two buckets must belong to the same account and be located in the same region.

2.1.1.2. Manage objects

2.1.1.2.1. Upload objects

Objects are the basic unit for data storage in Object Storage Service (OSS). Objects are also known as files. You can choose an upload method based on the size of the object to upload and your network environment.

OSS provides the following upload methods:

- Simple upload includes streaming upload and object upload. You can use this method to upload an object up to 5 GB in size.
- Form upload: supports the upload of an object up to 5 GB in size.
- Append upload: supports the upload of an object up to 5 GB in size.
- Resumable upload: supports concurrent and resumable upload of an object up to 48.8 TB in size. This method is suitable for the upload of large objects. You can use this method to upload an object up to 48.8 TB in size.
- Multipart upload: supports the upload of an object up to 48.8 TB in size. This method is suitable for the upload of large objects.

During object upload, you can configure object metadata and view upload progress in the Upload Tasks panel. After the object is uploaded, you can perform upload callback.

2.1.1.2.2. ACL

Object Storage Service (OSS) allows you to configure access control lists (ACLs) for objects to control access to the objects.

- You can configure ACL for an object when or after you upload the object. By default, if you do not specify ACL for an object, the ACL of the object is Inherited from Bucket.
- Inherited from Bucket: The ACL for the object is the same as that for the bucket.
- Private: Only the bucket owner or authorized users can read from and write to the objects in the bucket. Other users, including anonymous users, cannot access objects in the bucket.
- Public Read: Only the bucket owner or authorized users can read from and write to objects. Other users, including anonymous users, can only read from objects in the bucket.
- Public Read/Write: All users, including anonymous users, can perform read and write operations on objects in the bucket. The bucket owner are charged fees incurred by these operations. Therefore, we recommend that you use this ACL policy only when necessary.

2.1.1.2.3. Download objects

Object Storage Service (OSS) provides a variety of object download methods that you can choose to download objects stored in buckets based on your requirements.

OSS provides the following object download methods:

- Download objects to local disks: You can download objects stored in buckets to your local disks.
- Streaming download: If you want to download a large object or it takes a long time to download an object at a time, you can use streaming download to download the object incrementally until the entire object is downloaded.
- Range download: If you need only part of the data in an object, you can use range download to download data within the specified range.
- Resumable download: You may fail to download a large object if the network is unstable or other exceptions occur. In some cases, you may still fail to
 download the object even after multiple attempts. To handle this issue, OSS provides the resumable download feature. In resumable download,
 objects that you want to download are split into multiple parts and downloaded separately. After all parts are downloaded, these parts are combined
 into a complete object.
- Conditional download: You can specify one or multiple conditions when you download objects. If the specified conditions are met, the object is downloaded. If the specified conditions are not met, an error is returned and the object is not downloaded.

2.1.1.2.4. Search for objects

If a large number of objects are stored in your buckets, you can search for an object by specifying the prefix that the object name contains.

Usage notes

Search rules

You can search for objects by prefix. The string used to search for objects is case-sensitive and cannot contain forward slashes (/).

Search results

When you specify a prefix to search for an object in the root directory or a specified directory of a bucket, only the objects or subdirectories whose names contain the specified prefix are returned. Objects in subdirectories cannot be returned.

Examples

- Search for specific objects or directories within the root directory of the bucket
- Specify a prefix to search for specific objects or directories. Then, objects and directories that match the prefix within the root directory of the bucket are returned.

The following example shows the search result when you specify Example as the prefix to search for objects and directories within the root directory of the bucket named TestBucket.



Search for specific objects or subdirectories within a directory of the bucket

Select the directory and specify a prefix. Then, objects and subdirectories that match the prefix within the directory are returned.

The following example shows the search result when you specify Project as the prefix to search for objects and subdirectories within the directory named Examplesrcfolder1.



2.1.1.2.5. Manage objects by using directories

Object Storage Service (OSS) uses a flat structure instead of a hierarchical structure used by traditional file systems to store objects. All data in OSS are stored as objects in buckets. You can create simulated directories in OSS to help you categorize objects and control access to your objects in a simplified manner.

Structure

OSS uses objects whose names end with a forward slash (/) to simulate directories. The following example shows the structure of a bucket named examplebucket:



In the preceding structure:

- The following three objects have the log prefix in their names: log/date1.txt, log/date2.txt, and log/date3.txt. In the OSS console, a directory named log is displayed. Three objects named date1.txt, date2.txt, and date3.txt are stored within the directory.
- The destfolder/2021/photo.jpg object has the destfolder prefix in its name. In the OSS console, a directory named destfolder is displayed, which contains a subdirectory named 2021. An object named photo.jpg is stored in the 2021 subdirectory.

Access control based on directories

The following examples show how to grant third-party users different permissions to access the directories and objects in examplebucket described in the preceding section:

- The following objects within the log directory store the OSS access logs of a user in the last three days: log/date1.txt, log/date2.txt, and log/date3.txt. For support professionals to troubleshoot issues, such as slow access and object upload failures reported by the user, they need to view the logs stored in the three objects. In this case you can configure bucket policies to authorize other users to access your OSS resources
- An object named destfolder/2021/photo.jpg in examplebucket is a group photo of all your employees, which was taken on a 2021 spring outing. You want all your employees to have access to the object. In this case, you can set the ACL of the object to public read.

Implementation methods

You can create a directory in the OSS console. After you create a directory, you can upload objects to the directory.

Directories cannot be created or deleted by calling API operations. However, you can use OSS SDKs for various programming languages to create or delete directories by using the following methods:

- When you upload an object to OSS, you can add a directory name that ends with a forward slash (/) to the object name (key) to create a directory for the object. For example, when you upload a local file named localfile.txt to a bucket named examplebucket, you can set the name of the uploaded object to destfolder/localfile.txt. In this case, a directory named destfolder is created in examplebucket, and the uploaded object named localfile.txt is stored in destfolder. In this example, the destfolder directory is simulated by an object whose name is destfolder/ and whose size is 0.
- When you delete objects, you can specify a prefix that is contained in the names of all objects you want to delete. In this case, the directory whose name is the specified prefix and all objects within the directory are deleted. For example, if you specify a prefix "log", the directory named log and all objects within the directory are deleted.

2.1.1.2.6. Object tagging

Object tags can be used to classify objects. You can configure lifecycle rules and ACLs for objects based on their tags.

Usage notes

The object tagging feature uses a key-value pair to identify an object. You can add tags to objects when and after you upload objects.

- A maximum of 10 tags can be configured for each object. Tags associated with an object must have unique tag keys.
- A tag key can be up to 128 bytes in length. A tag value can be up to 256 bytes in length.
- Tag keys and tag values are case-sensitive.
- The key and the value of a tag can contain letters, digits, spaces, and the following special characters:
 - + = . : /
- Only the bucket owner and authorized users have read and write permissions on object tags. These permissions are independent of object access control lists (ACLs).
- In cross-region replication (CRR), object tags are replicated to the destination bucket.

Configure lifecycle rules for objects with the same tags

When you configure lifecycle rules, you can configure conditions for lifecycle rules to select subsets of objects to which the rules apply. You can configure conditions based on the object name prefixes, object tags, or both.

- If you configure conditions based on tags in one lifecycle rule, the rule applies only to objects that meet both the tag key and value conditions. • If you configure object name prefixes and multiple object tags in one lifecycle rule, the rule applies only to objects that match the object name prefixes and object tags.
- Example[.]

umpie:
LifecycleConfiguration>
<rule></rule>
<id>r1</id>
<prefix>rule1</prefix>
<tag><key>xx</key><value>1</value></tag>
<tag><key>yy</key><value>2</value></tag>
<status>Enabled</status>
<expiration></expiration>
<days>30</days>
<rule></rule>
<id>r2</id>
<prefix>rule2</prefix>
<tag><key>xx</key><value>1</value></tag>
<status>Enabled</status>
<transition></transition>
<days>60</days>
<storageclass>Archive</storageclass>
/LifecycleConfiguration>

In the preceding rules:

- Objects whose names are prefixed with rule1 and whose tagging configurations are xx=1 and yy=2 are deleted after the objects are stored for 30 days.
- The storage class of objects whose names are prefixed with rule2 and whose tagging configurations are xx=1 is converted to Archive after the objects are stored for 60 days.

Use RAM policies to manage permissions on objects with specified tags

You can authorize RAM users to manage object tags. You can also authorize RAM users to manage objects that have specific tags.

- Authorize RAM users to manage object tags
- You can authorize RAM users to manage all object tags or manage only specific object tags. If User A is authorized to set object tagging to allow=yes, this user can add the tagging configuration of allow=yes to objects. The following code provides an example on how to configure the corresponding RAM policy:



① Important After the RAM user is authorized to configure a specified tag for objects, the user can configure the tag only for existing objects. However, the user cannot configure the tag for objects when the user uploads the objects.

Authorize RAM users to manage objects that have specific tags You can authorize RAM users to manage all objects that have specific tags. For example, you can authorize User A to access all objects that have the tagging configuration of allow=yes. The following code provides an example on how to configure the corresponding RAM policy:
2.1.1.3. Data security

2.1.1.3.1. Erasure coding

Erasure Coding (EC) is a data storage mode used by Object Storage Service (OSS). Compared with triplicate storage, EC can provide higher data reliability at lower data redundancy levels.

EC

- EC involves the following two concepts:
- Data fragments (m): Data is divided into m data fragments.
- Parity fragments (n): n parity fragments are computed based on the m data fragments.

The m data fragments and n parity fragments located on different servers compose an erasure coding group. If the number of lost data fragments is equal to or less than n, the lost segments can be recovered based on the erasure coding algorithm. We recommend that you configure the value of m and n based on the number of servers.

- If you have 6 to 13 servers, we recommend that you set the values of both m and n to 2.
- If you have more than 14 servers, we recommend that you set the value of m to 8 and the value of n to 3.

Triplicate

Apsara Stack uses a flat design in which a linear address space is divided into slices called chunks. Each chunk is replicated into three copies stored on different data nodes of the storage cluster to ensure data reliability.



Triplicate storage involves three types of key component: the master, chunk server, and client. Chunk servers are data nodes where chunk copies are stored. Each write operation is executed by the client in the following manner:

- 1. The client receives your write request and determines the chunk that corresponds to the write operation by computing.
- 2. The client queries the master to find the chunk servers where the three copies of the chunk are stored.
- 3. The client sends write requests to the chunk servers returned from the master.
- 4. If the write operation succeeds on all three chunk copies, the client returns a success. Otherwise, the client returns a failure.

The master ensures that the copies of each chunk are distributed on different chunk servers across different racks. This prevents data unavailability caused by the failure of a single chunk server or rack. The distribution strategy of the master takes many factors of the storage system into account, such as chunk server disk usage, chunk server distribution across racks, power distribution conditions, and node workloads.

Comparison between EC and triplicate storage

Compared with triplicate storage, EC is a better solution in terms of storage usage and data reliability.

Item	EC	Triplicate storage
Storage usage	$_{m/~(m+n)}$. For example, the storage usage in EC storage of the 8+3 configuration can be calculated in the following method: $8/(8+3)\!=\!72.7\%$	1/3=33.3%
Reliability	Handles the loss of up to n fragments. Failures on up to n servers can be handled in the worst case. For example, when m is 8 and n is 3, failures on up to three servers can be handled.	Handles the loss of up to two replicas. Failures on up to two servers can be handled in the worst case.

2.1.1.3.2. Retention policies

Object Storage Service (OSS) supports the Write Once Read Many (WORM) feature. The feature helps prevent objects from being deleted or overwritten within a specified period of time. This complies with the regulations of the U.S. Securities and Exchange Commission (SEC) and Financial Industry Regulatory Authority. (FINRA).

Prerequisites

- Retention policies are not supported in the China (Guangzhou), China (Nanjing Local Region), and US (Virginia) regions.
- Versioning is not enabled for the bucket for which you want to configure retention policies. For more information about versioning, see Overview.

Scenarios

OSS provides strong compliance policies. You can configure time-based retention policies for OSS buckets. After you configure and lock a retention policy for a bucket, you can upload objects to or read objects from the bucket. However, you cannot delete objects from the bucket or the retention policy within the retention period. You can delete the objects only after the retention period expires. The WORM feature is suitable for fields that involve privacy, such as finance, insurance, health care, securities, and logs data classified protection censorship. OSS enables you to build compliant buckets on the cloud.

? Note

OSS is accredited by Cohasset Associates in audit and meets specific requirements for electronic data storage. OSS buckets that are configured with retention policies can be used for business that is subject to regulations such as SEC Rule 17a-4(f), CFTC Rule 1.31(c)-(d), and FINRA Rule 4511(c). For more information, see OSS Cohasset Assessment Report.

Limits

- You can configure retention policies only for buckets in OSS.
- We recommend that you do not enable the OSS-HDFS service and configure retention policies for a bucket at the same time.
- Assume that OSS-HDFS is enabled for a bucket for which a retention policy is configured. When you use methods supported by OSS-HDFS to delete an object from the __dlsdata/ directory, you receive a message that indicates the object is deleted. However, OSS actually retains the object if the deletion occurs within the retention period and cannot recognize and delete the object after the retention period ends.
- During the retention period, you can configure lifecycle rules to convert the storage classes of objects in the bucket. This way, you can reduce costs and ensure compliance. For more information, see Lifecycle rules based on the last modified time.

Rules

You can configure only a single time-based retention policy for each bucket. The policy specifies a retention period that ranges from one day to 70 years.

For example, assume that you created a bucket named examplebucket on June 1, 2013, and uploaded the file1.txt, file2.txt, and file3.txt objects to the bucket at different points in time. On July 1, 2014, you created a retention policy for the bucket and specified a five-year retention period. The following table describes the upload and expiration dates of the objects.

Object	Upload date	Expiration date
file1.txt	June 1, 2013	May 31, 2018
file2.txt	July 1, 2014	June 30, 2019
file3.txt	September 30, 2018	September 29, 2023

- Implementation rules
 - By default, a time-based retention policy is in the InProgress state after the policy is created for a bucket. The InProgress state lasts for 24 hours. The retention policy protects the resources in the bucket within 24 hours after the policy is created.
 - Within 24 hours after the retention policy is created: If the retention policy is not locked, the bucket owner and authorized users can delete the
 policy. If the retention policy is locked, the retention period of the policy cannot be shortened and the policy cannot be deleted. You can only
 extend the retention period.
- 24 hours after the retention policy is created: If the retention policy is not locked, the policy becomes invalid.
- If you attempt to delete or modify data in the protected bucket, a 409 FileImmutable error is returned.
- Deletion rules
 - A time-based retention policy is a metadata attribute of a bucket. If a bucket is deleted, the retention policy and the access control list (ACL) of the bucket are also deleted. If a bucket is empty, the bucket owner can delete the bucket. This way, the retention policy of the bucket is deleted.
 - If the retention policy is not locked within 24 hours after the policy is created, the bucket owner and authorized users can delete the policy.
 - If a bucket contains objects that are protected within the retention period, you cannot delete the bucket or the retention policy.

Use the OSS console

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket in which you want to create a directory.
- 3. In the left-side navigation pane, choose Basic Settings > Retention Policy. In the Retention Policy section, click Configure.
- 4. Click Create Policy.
- 5. In the Create Policy dialog box, set Retention Period.

The retention period ranges from one day to 70 years.

- 6. Click **OK**.
 - After you create the policy, the policy is in the InProgress state. You can click Lock or Delete to lock or delete a policy in the InProgress state.
- 7. Click Lock.
- 8. In the message that appears, click **OK**.

() Important

- The policy enters the Locked state. You cannot delete the policy or shorten the retention period. However, you can click **Edit** to extend the retention period.
- During the retention period, data in the bucket is protected. If you attempt to delete or modify the data, the following error message is displayed: The file is locked and cannot be operated.

Use OSS SDKs

Overview.

The following sample code provides an example on how to configure retention policies by using OSS SDKs for common programming languages. For more information about the sample code that is used to configure retention policies by using OSS SDKs for other programming languages, see

```
Reliebis
import com.aliyun.oss.ClientException;
import com.aliyun.oss.OSS;
import com.aliyun.oss.OSSClientBuilder;
import com.aliyun.oss.OSSException;
import com.alivun.oss.model.InitiateBucketWormReguest;
import com.aliyun.oss.model.InitiateBucketWormResult;
public class Demo {
    public static void main(String[] args) throws Exception {
         // In this example, the endpoint of the China (Hangzhou) region is used. Specify your actual endpoint.
         String endpoint = "https://oss-cn-hangzhou.aliyuncs.com";
         // The AccessKey pair of an Alibaba Cloud account has permissions on all API operations. Using these credentials to perform operations
in OSS is a high-risk operation. We recommend that you use a Resource Access Management (RAM) user to call API operations or perform routine O
&M. To create a RAM user, log on to the RAM console.
    String accessKeyId = "yourAccessKeyId";
         String accessKeySecret = "yourAccessKeySecret";
         // Specify the name of the bucket. Example: examplebucket.
        String bucketName = "examplebucket";
         // Create an OSSClient instance.
         OSS ossClient = new OSSClientBuilder().build(endpoint, accessKeyId, accessKeySecret);
         try (
             // Create an InitiateBucketWormRequest request.
             InitiateBucketWormRequest initiateBucketWormRequest = new InitiateBucketWormRequest(bucketName);
             // Set the retention period to one day.
             initiateBucketWormRequest.setRetentionPeriodInDays(1);
             // Create a retention policy.
             InitiateBucketWormResult initiateBucketWormResult = ossClient.initiateBucketWorm(initiateBucketWormRequest);
             // Display the ID of the retention policy.
             String wormId = initiateBucketWormResult.getWormId();
             System.out.println(wormId);
         } catch (OSSException oe) {
             System.out.println("Caught an OSSException, which means your request made it to OSS, "
                      + "but was rejected with an error response for some reason.");
             System.out.println("Error Message:" + oe.getErrorMessage());
System.out.println("Error Code:" + oe.getErrorCode());
System.out.println("Request ID:" + oe.getRequestId());
             System.out.println("Host ID:" + oe.getHostId());
         } catch (ClientException ce) {
             System.out.println("Caught an ClientException, which means the client encountered "
                     + "a serious internal problem while trying to communicate with OSS, "
                     + "such as not being able to access the network.");
             System.out.println("Error Message:" + ce.getMessage());
         } finally {
             if (ossClient != null) {
                 ossClient.shutdown();
             }
        }
    }
}
```

Use ossutil

For more information about how to configure retention policies by using ossutil, see worm (manage retention policies).

Use the RESTful API

If your program requires more custom options to configure retention policies, you can call RESTful API operations. To directly call an API, you must include the signature calculation in your code. For more information, see InitiateBucketWorm.

FAQ

• Q: What are the benefits of a retention policy?

A: A retention policy can be used to meet data security standards. Within the retention period of a retention policy, data cannot be modified or deleted. The data that is protected by using RAM policies and bucket policies can be modified and deleted.

• Q: What are the scenarios in which a retention policy can be used?

A: You can use a retention policy if you want to store important data that is infrequently accessed and not allowed to be modified or deleted. This type of data includes medical records, technical documents, and contracts. You can store these objects in a specific bucket and configure a retention policy for the bucket.

- Q: Can I configure a retention policy at the object level?
- A: No, retention policies can be configured only at the bucket level.
- Q: How do I calculate the expiration time of an object within a retention period?

A: The expiration time of an object within a retention period can be calculated based on the retention period and the time when the object is last modified. For example, the retention period of Bucket A is 10 days. An object in the bucket is last modified at 12:00 on February 15, 2022. In this case, the object expires at 12:01 on February 25, 2022.

- Q: How do I delete a bucket that is protected by a retention policy?
 - A: If the bucket contains no objects, you can delete the bucket. This way, the retention policy is deleted.
 - If the bucket contains objects, the bucket cannot be deleted even if the retention period ends. In this case, you can delete all objects from the bucket and then delete the bucket.
- If the bucket contains objects that are protected within the retention period, the bucket cannot be deleted.
- Q: Are objects that are protected within the retention period of a retention policy retained if my account has overdue OSS payments?
- A: If your account has overdue payments, Alibaba Cloud retains data based on the corresponding terms and conditions of your contract.
- Q: Can an authorized RAM user configure a retention policy?

A: All API operations related to the retention policy are available and they support RAM policies. RAM users that are granted permissions by using RAM policies can create or delete a retention policy by using the OSS console, API operations, or OSS SDKs.

2.1.1.3.3. Resource isolation

Object Storage Service (OSS) slices user data and discretely stores the sliced data in a distributed file system based on specific rules. The user data and its indexes are stored separately.

OSS uses symmetric AccessKey pairs to authenticate users and verifies the signature in each HTTP request sent by users. If verification is successful, OSS reassembles the distributed data. This way, OSS implements data storage isolation between different tenants.

2.1.1.3.4. Disaster recovery

Object Storage Service (OSS) provides multiple disaster recovery capabilities to ensure data security and improve availability.

To ensure data availability, OSS provides the following disaster recovery capabilities.

Capability	Description
Zone-disaster recovery	Zone-disaster recovery allows you to store multiple replicas of your data in multiple zones within the same region. This feature protects your data from being lost and helps you recover your business if a single zone fails.
Cross-region replication (CRR)	CRR enables the automatic and asynchronous (near real-time) replication of objects across OSS buckets in different regions. Operations such as the creation, overwriting, and deletion of objects can be synchronized from a source bucket to a destination bucket.
Three data centers across two regions	If your business has high requirements on data backup, you can use zone-disaster recovery and cross-region replication to build a disaster recovery solution based on three data centers across two regions.

2.1.1.3.5. Access permissions and account authorization

By default, the access control list (ACL) of Object Storage Service (OSS) resources, including buckets and objects, is set to private to ensure data security. Only the bucket owner and authorized users can access these resources. OSS allows you to configure a variety of policies to grant third-party users specific permissions to access or use your OSS resources.

OSS provides the following access permission policies

Policy	Description
RAM Policy	Resource Access Management (RAM) is a service provided by Alibaba Cloud to manage access permissions on resources. RAM policies are configured based on users. You can manage users by configuring RAM policies. For users such as employees, systems, or applications, you can control which resources are accessible. For example, you can create a RAM policy to grant users only read permissions on a bucket.
Bucket Policy	A bucket policy is a resource-based authorization policy. Compared with RAM policies, bucket policies can be easily configured by using GUI in the console. In addition, the owner of a bucket can configure bucket policies for the bucket without RAM permissions. You can configure bucket policies to grant permissions to the RAM users of other Apsara Stack accounts or anonymous users who access OSS by using the specified IP addresses.
Bucket ACL	You can configure the ACL of a bucket when you create the bucket or modify the ACL of a created bucket. Only the owner of a bucket can configure or modify the ACL of the bucket. You can set the ACL of a bucket to one of the following values: Public Read/Write, Public-Read, and Private.
Object ACL	You can also configure the ACL of each object stored in OSS. You can configure the ACL of an object when you upload the object or modify the ACL of an uploaded object based on your requirements. You can set the ACL of an object to one of the following values: Inherited from bucket, Public Read/Write, Public-Read, and Private.
Hotlink protection	You can configure a Referer whitelist for a bucket to prevent your resources in the bucket from unauthorized access.
CORS	Cross-origin resource sharing (CORS) is a standard cross-origin solution provided by HTML5 to allow web application servers to control cross-origin access, which ensures the security of data transmission across origins.

2.1.1.3.6. Server-side encryption

Object Storage Service (OSS) supports server-side encryption. When you upload an object to a bucket for which server-side encryption is enabled, OSS encrypts the object and stores the encrypted object. When you download the encrypted object from OSS, OSS automatically decrypts the object and returns the decrypted object to you. In addition, a header is added in the response to indicate that the object is encrypted on the OSS server.

Encryption methods

OSS protects static data by using server-side encryption. You can use this method in scenarios in which additional security or compliance is required, such as the storage of deep learning samples and online collaborative documents.

You can use SSE-OSS to encrypt all your objects. To improve security, OSS uses master keys that are rotated on a regular basis to encrypt data keys. You can use this method to encrypt and decrypt multiple objects at a time.

Server-side encryption by using OSS-managed keys

OSS generates and manages data keys used to encrypt data, and provides strong and multi-factor security measures to protect data. OSS server-side encryption uses AES-256, which is one of the advanced encryption standard algorithms, and SM4, which is one of the Chinese cryptographic algorithms.

Use the following configuration methods:

- · Configure the default server-side encryption method for a bucket
- Set the default encryption method to SSE-OSS and specify the encryption algorithm as AES-256 . This way, all objects uploaded to this bucket are encrypted.
- Configure an encryption method for a specific object

When you upload an object or modify the metadata of an object, include the x-oss-server-side-encryption parameter in the request and set the parameter value to AES-265 or SM4. This way, the object is encrypted by using SSE-OSS.

2.1.1.3.7. Client-side encryption

Client-side encryption is performed to encrypt objects on the local client before the objects are uploaded to Object Storage Service (OSS).

Disclaimer

- When you use client-side encryption, you must ensure the integrity and validity of the customer master key (CMK). If the CMK is incorrectly used or lost due to improper key management, you are responsible for all losses and consequences caused by decryption failures.
- When you copy or migrate encrypted data, you are responsible for the integrity and validity of the object metadata related to client-side encryption. If the encrypted metadata is incorrectly used or lost due to improper maintenance, you are responsible for all losses and consequences caused by decryption failures.

Encryption

In client-side encryption, a random data key is generated for each object to perform symmetric encryption on the object. The client uses a CMK to generate a random data encryption key. The encrypted data key is uploaded as a part of the object metadata and stored in the OSS server. When an encrypted object is downloaded, the client uses the CMK to decrypt the random data key and then uses the data key to decrypt the data of the object. The CMK is used only on the client and is not transmitted over the network or stored in the server, which ensures data security.

Use customer-managed CMK

To use this method for client-side encryption, you must generate and manage CMKs by yourself. When you implement client-side encryption on an object to upload, you must upload a symmetric or asymmetric CMK to the client. The following figure shows the specific encryption process.



User environment

- Encrypt and upload an object
- i. You must provide the client with a symmetric or asymmetric CMK.
- ii. The client uses the CMK to generate a one-time symmetric data key that is used only to encrypt the current object to upload. The client generates a random and unique data key for each object to upload.
- iii. The client uses the data key to encrypt the object to upload and uses the CMK to encrypt the data key.
- iv. The encrypted data key is included in the metadata of the uploaded object in OSS.
- Download and decrypt an object
- i. The client downloads an encrypted object. The encrypted data key is included in the metadata of the object.
- ii. The client uses the object metadata information to determine the CMK used to generate the data key and uses this CMK to decrypt the encrypted data key. Then, the client uses the decrypted data key to decrypt the object.

() Important

- CMKs and unencrypted data are not sent to OSS. Therefore, keep your CMKs secure. If a CMK is lost, objects encrypted by using the data keys generated by this CMK cannot be decrypted.
- Data keys are randomly generated by the client.

Usage notes

- To perform client-side encryption on objects that are larger than 5 GB in size before you upload the objects, you must use multipart upload. When you use multipart upload to upload an object, you must specify the total size of the object and the size of each part. The size of each part except for the last part must be the same and be a multiple of 16 bytes.
- After you upload objects encrypted on the client, object metadata related to client-side encryption is protected. In this case, CopyObject cannot be called to modify object metadata.

2.1.1.3.8. Versioning

OSS allows you to configure versioning for a bucket to protect objects that are stored in the bucket. After you enable versioning for a bucket, data that is overwritten or deleted in the bucket is saved as a previous version. After you configure versioning for a bucket, you can recover objects in the bucket to any previous version to protect your data from being accidentally overwritten or deleted.

Versioning states

A bucket can be in one of the following versioning states: disabled, enabled, and suspended.

- By default, versioning is disabled for a bucket. After versioning is enabled for a bucket, the versioning state of the bucket cannot be set back to disabled. However, you can suspend versioning for a bucket that has versioning enabled.
- When an object is uploaded to a bucket for which versioning is enabled, OSS generates a random string as the globally unique version ID of the
 object.
- When an object is uploaded to a bucket for which versioning is suspended, OSS generates the "null" string as the version ID of the object.

Scenarios

To ensure data security, we recommend that you use versioning in the following scenarios:

Recover deleted data

- You can configure versioning to recover deleted data.
- Recover overwritten data
- Numerous temporary versions are created in scenarios where modifications are frequently made, such as online collaborative documents and documents stored in online storage. You can use the versioning feature to retrieve a specified version of an object stored in the bucket.

Data protection

The following table describes how OSS processes deleted and overwritten data in buckets in different versioning states to help you understand the data protection mechanism of versioning.

Versioning state	Object overwriting	Object deletion
Disabled	The existing object is overwritten and cannot be recovered. Only the current object version is retained in the bucket.	The object is deleted and cannot be accessed.
Enabled	A new version with a unique ID is generated for the object. The existing object is stored as a previous version.	A delete marker with a globally unique version ID is added to the object as the current version. The existing object is stored as a previous version.
Suspended	A new version whose version ID is null is generated for the object. If the object already has a previous version or delete marker whose version ID is null, the previous version or delete marker is overwritten by the new null version. Other previous versions or delete markers whose version IDs are not null are not affected.	A delete marker whose version ID is null is added to the object. If the object already has a previous version or delete marker whose version ID is null, the previous version or delete marker is overwrithen by the new delete marker. Other previous versions or delete markers whose version IDs are not null are not affected.

The following examples provide figures to show how OSS processes data when an object with the same name as an existing object is uploaded to or an object is deleted from a bucket for which versioning is enabled or suspended. All version IDs in the figures are in the simple format for readability.

- Overwrite an object in a versioning-enabled bucket
- When you upload an object repeatedly to a versioning-enabled bucket, the object is overwritten in each upload. A version with a unique version ID is generated for the object each time when the object is overwritten.

Upload the object for the third time after versioning is enabled.



· Delete an object from a versioning-enabled bucket

When you delete an object from a versioning-enabled bucket, OSS adds a delete marker to the object as the current version of the object instead of permanently deleting this object. The previous versions of the object are not deleted. If you upload an object with the same name after the delete marker is added, a new version with a unique version ID is added as the current version.



Overwrite an object in a versioning-suspended bucket

When you upload an object with the same name as an existing object in a bucket for which versioning is suspended, a new version whose version ID is null is added and the previous versions of the object are retained. If you upload another object with the same name to the bucket again, a new version whose version ID is null overwrites the current version whose version ID is null.



Delete an object from a versioning-suspended bucket

When you delete an object from a bucket for which versioning is suspended, OSS adds a delete marker to the object as the current version of the object instead of permanently deleting this object. The previous versions of the object are not deleted.



In a versioning-enabled bucket, deleted and overwritten data is stored as previous versions. After you enable versioning for a bucket, you can recover objects in the bucket to a previous version to protect your data from being accidentally overwritten or deleted.

2.1.1.4. Data processing

2.1.1.4.1. IMG

You can add Image Processing (IMG) parameters to GetObject requests to process image objects stored in Object Storage Service (OSS). For example, you can add image watermarks to images or convert image formats.

OSS allows you to use one or more parameters to process images, or encapsulate multiple IMG parameters in a style to process images. When multiple IMG parameters are included in a request, OSS processes the image in the order of the parameters.

You can use object URLs, API operations, and OSS SDKs to process images. The following table describes the IMG operations supported by OSS.

IMG operation	Parameter	Description
Resize images	resize	Resizes images to a specified size.
Incircle	circle	Crops images based on the center point of images to rounds of a specified size.
Custom crop	crop	Crops images to rectangles of a specified size.
Indexed cut	indexcrop	Cuts images along a specified horizontal or vertical axis and selects one of the cut images.
Rounded rectangle	rounded-corners	Crops images to rounded rectangles based on the specified rounded corner size.
Automatic rotation	auto-orient	Auto-rotates images for which the auto-orient parameter is configured.
Rotate	rotate	Rotates images clockwise at a specified angle.
Blur	blur	Blurs images.
Adjust brightness	bright	Adjusts the brightness of images.
Sharpen	sharpen	Sharpens images.
Adjust contrast	contrast	Adjusts the contrast of images.

Gradual display	interlace	Configures gradual display for the JPG images.
Adjust image quality	quality	Adjusts the quality of JPG and WebP images.
Convert format	format	Converts image formats.
Add watermarks	watermark	Adds image or text watermarks to images.
Query average tone	average-hue	Queries the average tone of images.
Query image information	info	Queries image information, including basic information and Exchangeable Image File Format (EXIF) information.

2.1.1.4.2. Video snapshots

This topic describes the parameters that you can configure to capture video snapshots and provides examples.

Usage notes

- When you capture video snapshots, you are charged based on the number of captured images.
- Object Storage Service (OSS) can capture images from video objects only in the H.264 and H.265 formats.
- By default, OSS does not automatically store captured images. You must manually download the captured images to your local storage devices.

Parameters

Operation type: video

Operation name: snapshot

Paramete r	Description	Valid value
t	Specifies the time when the image is to be captured.	[0, video duration] Unit: ms
w	Specifies the width based on which to capture the image. If this parameter is set to 0, the width of the image to capture is automatically calculated.	[0, video width] Unit: px
h	Specifies the height based on which to capture the image. If this parameter is set to 0, the height of the image to capture is automatically calculated. If both w and h are set to 0, the width and height of the source image are used as those of image to capture.	[0, video height] Unit: px
m	Specifies the mode used to capture the image. If this parameter is not specified, the image is captured in the default mode. In other words, the image at the specified point of time in the video is captured. If this parameter is set to fast, the most recent key frame before the specified time is captured.	fast
f	Specifies the format of the captured image.	jpg and png
ar	Specifies whether to automatically rotate the image based on the video information. If this parameter is set to auto, the system automatically rotates the image based on the video information.	auto

Examples

• Use the fast mode to capture the image at the seventh second of the video. Export the captured image as a JPG image with the width of 800 px and height of 600 px.

The URL of the processed image is in the following format: <Source video URL>?x-oss-process=video/snapshot,t_7000,f_jpg,w_800,h_600,m_fast -

• Use the default mode to capture the image at the fiftieth second of the video accurately. Export it as a JPG image with the width of 800 px and height of 600 px.

The URL of the processed image is in the following format: <Source video URL>?x-oss-process=video/snapshot,t_50000,f_jpg,w_800,h_600

2.2. Usage notes

Before you use OSS, you must understand the following content:

To allow other users to use all or part of OSS features, you must create RAM users and grant permissions to the users by configuring RAM policies. Before you use OSS, you must also understand the following limits.

Item	Limit
Bucket	You can create up to 100 buckets.After a bucket is created, its name and region cannot be modified.
Upload objects	 Objects larger than 5 GB cannot be uploaded by using the following modes: console upload, simple upload, form upload, or append upload. To upload an object that is larger than 5 GB, you must use multipart upload. The size of an object uploaded by using multipart upload cannot exceed 48.8 TB. If you upload an object that has the same name of an existing object in OSS, the new object will overwrite the existing object.
Delete objects	 Deleted objects cannot be recovered. You can delete up to 100 objects at a time in the OSS console. To delete more than 100 objects at a time, you must call an API operation or use an SDK.
Lifecycle	You can configure up to 1,000 lifecycle rules for each bucket.

2.3. Quick start

2.3.1. Log on to the OSS console

This topic describes how to log on to the Object Storage Service (OSS) console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel.
- A web browser is available. We recommend that you use Google Chrome.

Procedure

- 1. Enter the URL of the Apsara Uni-manager Management Console in the address bar and press the Enter key.
 - ? Note
 - $\circ~$ You can click the current language in the upper-right corner to switch to another language.

2. Enter your username and password.

- Obtain the username and password from an operations administrator.
 - ? Note
 - First logon

The first time that you log on to the Apsara Uni-manager Management Console, you need to change the password of your account. The password must be 10 to 32 characters in length. It must contain all of the following character types: uppercase letters, lowercase letters, digits, and special characters. Supported special characters include: ! @ # \$ %

• Forget password

If you have forgotten your password, click Forgot Password. On the page that appears, enter the username of your account, the email address that was used to create the account, and the CAPTCHA code. Then, the system sends a link for resetting the password to the specified email address.

3. Click Log On.

- 4. If multi-factor authentication (MFA) is enabled for your account, perform the corresponding operations in the following scenarios:
 - You log on to the Apsara Uni-manager Management Console for the first time after MFA is enabled by the administrator.
 - a. On the Bind Virtual MFA Device page, bind an MFA device.
 - b. Enter the username and password again as in Step 2 and click $\mbox{Log On}.$
 - c. Enter a six-digit MFA verification code and click Authenticate.
 - You have enabled MFA and bound an MFA device.

Enter a six-digit MFA verification code and click Authenticate.

⑦ Note
For information about how to bind and enable MFA, see Manage MFA in Apsara Uni-manager Management Console User Guide

5. In the top navigation bar, choose **Products > Object Storage Service**.

2.3.2. Create a bucket

OSS stores files as objects in buckets. When you upload files to OSS, you must create a bucket or select an existing bucket. This topic describes how to create a bucket.

- 1. Log on to the OSS console. For more information, see Log on to the OSS console
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click Create Bucket.
- 3. In the **Create Bucket** panel, set the required parameters.
- The following table describes the parameters.

Parameter	Description	
Organization Name	Select an organization from the drop-down list for the bucket.	
Resource Set Name	Select a resource set from the drop-down list for the bucket.	
Region	Select a region from the drop-down list for the bucket. Image: The region of a bucket cannot be changed after the bucket is created. Image: Image: The region of a bucket cannot be changed after the bucket is created. Image: Image: Image: The region of a bucket cannot be changed after the bucket is created. Image: Image: Image: Image: The region of a bucket cannot be changed after the bucket is created. Image: Imag	
Cluster	Select a cluster for the bucket.	

	Enter the name of the bucket.
Bucket Name	 The bucket name must comply with the naming conventions. The bucket name must be globally unique among all existing buckets in OSS. The bucket name cannot be changed after the bucket is created.
Storage Class	Use the Standard storage class. Apsara Stack OSS supports only the Standard storage class.
Storage Quota	Specify the storage space of the bucket. Valid values: 1 to 2000000 TB or 1 to 2048000000 GB.
Versioning	 Select whether to enable versioning. Enable: If you enable versioning for a bucket, objects that are overwritten or deleted in the bucket are stored as previous versions. Versioning allows you to recover objects in a bucket to a previous version, and protects your data from being accidentally overwritten or deleted. Disable: If you do not enable versioning for a bucket, you cannot recover objects that are overwritten or deleted in the bucket.
Access Control List (ACL)	 Set the ACL of the bucket. Valid values: Private: Only the owner of the bucket and authorized users have read and write access to objects in the bucket. Other users, including anonymous users, do not have access to objects in the bucket. Public Read: Only the owner of the bucket has read and write access to objects in the bucket. Other users, including anonymous users, have only read access to objects in the bucket. Public Read/Write: All users, including anonymous users, have read and write access to objects in the bucket. Charges generated from these operations are paid by the owner of the bucket. Exercise caution when you set the ACL to Public Read/Write. Note You can change the ACL of a bucket after the bucket is created. For more information, seeModify bucket ACLs.
Encryption Method	 Configure server-side encryption for the bucket. Valid values: None: Server-side encryption is not configured. OSS-Managed: Keys managed by OSS are used to encrypt objects in the bucket.
Disable Zone-disaster Recovery	 Select whether to disable zone-disaster recovery for the bucket. If Disable Zone-disaster Recovery is set to Disable, zone-disaster recovery is enabled. Objects in the bucket are stored in zone-redundant storage (ZRS) mode. ZRS stores objects in three zones within the same region. You have access to objects in the bucket even if one of the zones fails due to an outage, fire accident, or other factors. If Disable Zone-disaster Recovery is set to Enable, zone-disaster recovery is disabled. Objects in the bucket are stored in locally redundant storage (LRS) mode. LRS stores the copies of each object across different devices within the same zone. This way, OSS ensures data reliability and availability even if two storage devices fail at the same time.

4. Click **OK**.

After the bucket is created, you can go to the **Overview** tab of the bucket details page to view information about the bucket, such as the organization, resource set, domain names, basic settings, and bound VPCs.

2.3.3. Upload objects

After you create a bucket, you can upload objects to it.

Prerequisites

A bucket is created. For more information, see Create buckets.

Background information

You can upload an object of any format to a bucket. You can use the OSS console to upload an object up to 5 GB in size. To upload an object larger than 5 GB, use OSS SDKs or call an API operation.

- 1. Log on to the OSS console.
- 2. Click **Buckets**. On the Buckets page that appears, click the name of the bucket to which you want to upload objects.
- 3. In the left-side navigation pane, click **Files**.
- 4. Click Upload.
- 5. In the **Upload** panel, set the parameters described in the following table.

Parameter	Description
Upload To	 Set the directory to which you want to upload the objects. Current: Objects are uploaded to the current directory. Specified: Objects are uploaded to the specified directory. You must enter the directory name. If the directory with the entered name does not exist, OSS automatically creates the directory and uploads the object to the directory.

File ACL	Select the access control list (ACL) of the objects that you want to upload. Default value: Inherited from Bucket. Valid values:			
	 Inherited from Bucket: The ACL of each object is the same as that of the bucket. This is the default ACL of uploaded objects. 			
	• Private : Only the owner or authorized users can read and write objects in the bucket. Other users, including anonymous users cannot access the objects in the bucket without authorization.			
	 Public Read: Only the bucket owner can perform write operations on objects in the bucket. Other users, including anonymous users, can perform only read operations on objects in the bucket. 			
	Public Read/Write: All users, including anonymous users, can read and write objects in the bucket.			
Upload	Drag one or more objects to upload to this section, or click Upload to select one or more objects to upload.			
	1 Important			
	 When the object to upload has the same name as an existing object in the bucket, the existing object is overwritten. 			
	 If you upload a directory, only the files in the directory are uploaded. The files are stored in the same directory in the bucket. 			
	 The name of an uploaded object must comply with the following conventions: 			
	The name must be encoded in UTF-8.			
	 The name is case-sensitive. 			
	The name must be 1 to 1,023 bytes in length.			
	 The name cannot start with a forward slash (/) or backslash (\). 			

6. In the Upload Tasks panel, wait until the upload task is completed. During the upload process, you can click Cancel All to cancel all upload tasks. After the task is completed, click Removed to remove the task.

① Important Do not refresh or close the Upload Tasks panel when objects are being uploaded. Otherwise, the upload tasks are interrupted.

2.3.4. Obtain object URLs

You can obtain the URL of an uploaded object and share the URL with other users to preview or download the object.

Prerequisites

An object is uploaded to the bucket. For more information, see Upload objects.

Procedure

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the Files tab.
- 4. Obtain object URLs
 - Obtain the URL of a single object.

Click the name of the object whose URL you want to obtain, or click View Details in the Actions column that corresponds to the object. In the View Details panel, click Copy File URL.

Batch export URL lists

Select the objects that you want to share. Choose Batch Operation > Export URL List.

2.4. Buckets

2.4.1. View bucket information

You can view the detailed information about created buckets in the Object Storage Service (OSS) console.

Prerequisites

A bucket is created. For more information, see Create buckets.

Procedure

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name to which you want to upload objects.
- 3. On the Overview tab, you can view the information about the bucket, which includes Organization and Resource Set, Domain Names, Basic Settings, and VPC

```
⑦ Note
```

- You can use domain names to manage a bucket over HTTP or HTTPS.
- You can view the storage quota of a bucket in the Basic Settings section. For more information, see Configure storage quota.

2.4.2. Delete buckets

You can delete a bucket in the OSS console.

Prerequisites

All objects and parts in the bucket are deleted. For more information, see Delete objects and Manage parts.

Marning Deleted objects, parts, and buckets cannot be recovered. Exercise caution when you delete objects, parts, and buckets.

1. Log on to the OSS console.

- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name to which you want to upload objects.
- 3. In the upper-right corner, click **Delete Bucket**.
- 4. In the message that appears, click **Confirm**.

2.4.3. Modify bucket ACLs

OSS provides access control list (ACL) to control access to buckets. By default, the ACL of a bucket is private when you create the bucket. You can modify the ACL of a bucket after the bucket is created.

Prerequisites

A bucket is created. For more information, see Create buckets.

Background information

- You can set the ACL of a bucket to one of the following values:
- Private: Only the owner or authorized users of this bucket can read and write objects in the bucket. Other users, including anonymous users cannot access the objects in the bucket without authorization.
- Public Read: Only the owner or authorized users of this bucket can write objects in the bucket. Other users, including anonymous users can only read
 objects in the bucket.
- Public Read/Write: Any users, including anonymous users, can read and write the objects in the bucket.

▲ Warning If you set the ACL of a bucket to Public Read or Public Read/Write, other users can read the data in the bucket without authentication, which may result in security risks. To ensure the security of your data, we recommend that you set the ACL of your bucket to private.

Procedure

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the Basic Settings tab. Find the Access Control List (ACL) section.
- 4. Click **Configure**. Modify the bucket ACL.
- 5. Click Save.

2.4.4. Configure static website hosting

You can configure static website hosting for a bucket in the Object Storage Service (OSS) console so that users can access the website by using the domain name of the bucket.

Prerequisites

A bucket is created. For more information, see Create buckets.

Procedure

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the **Basic Settings** tab. Find the **Static Pages** section.
- 4. Click Configure and then set the parameters described in the following table.

Parameter	Description
Default Homepage	Specify an index page that functions similar to index.html. Only HTML objects can be set to the index page. If you do not specify this parameter, static website hosting is disabled.
Default 404 Page	Set the default 404 page that is displayed when the requested resource does not exist. Only HTML, JPG, PNG, BMP, or WebP objects in the root folder of the bucket can be set to the default 404 page. If you do not specify this parameter, Default 404 Page is disabled.

5. Click Save.

2.4.5. Configure hotlink protection for a bucket

You can configure hotlink protection for a bucket in the Object Storage Service (OSS) console to prevent data in your bucket from being accessed by unauthorized domain names.

Prerequisites

A bucket is created. For more information, see Create buckets.

Background information

The hotlink protection feature allows you to configure a Referer whitelist for a bucket. This way, only requests from domain names included in the Referer whitelist can access your data in the bucket. You can configure Referer whitelists based on the Referer header field in HTTP and HTTPS requests.

After you configure a Referer whitelist for a bucket, OSS verifies requests to objects in the bucket only when the requests are initiated from signed URLs or anonymous users. Requests that contain the Authorization field in the header are not verified.

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the Basic Settings tab. Find the Hotlink Protection section.
- 4. Click **Configure** and configure the following parameters:
 - Enter domain names or IP addresses in the **Referer Whitelist** field. Separate multiple Referers by using line feed. You can use asterisks (*) and question marks (?) as wildcards. Examples:

- If you add www.example.com to the Referer whitelist, requests sent from URLs that start with www.example.com, such as www.example.com/123 and www.example.com.cn are allowed.
- If you add *www.example.com/ to the Referer whitelist, requests sent from http://www.example.com/ and https://www.example.com/ are allowed.
- You can use an asterisk (*) as a wildcard character to specify zero or multiple characters. For example, if you add *.example.com to the Referer whitelist, requests sent from URLs such as help.example.com and www.example.com are allowed.
- You can use a question mark (?) as a wildcard character to specify a single character. For example, if you add example?.com to the Referer
 whitelist, requests sent from URLs such as examplea.com and exampleb.com are allowed.
- You can add domain names or IP addresses that include a port number, such as www.example.com:8080 and 10.10.10.10.8080, to the Referer whitelist.
- Select whether to turn on Allow Empty Referer to allow requests in which the Referer field is empty.

An HTTP or HTTPS request with an empty Referer field indicates that the request does not contain the Referer field or the value of the Referer field is empty.

If you do not allow empty Referers fields, only HTTP or HTTPS requests which include an allowed Referer field can access the objects in the bucket.

(?) Note By default, if you use the bucket endpoint to preview an MP4 object, the browser sends a request that contains the Referer field and a request that does not contain the Referer field at the same time. Therefore, to allow access to the MP4 objects in your bucket, you must not only add the bucket endpoint to the Referer whitelist but also allow empty Referer fields. To preview a non-MP4 object by using the bucket domain name, you need only to allow empty Referer fields.

5. Click Save.

2.4.6. Configure CORS

You can configure cross-origin resource sharing (CORS) in the Object Storage Service (OSS) console to enable cross-origin access.

Prerequisites

A bucket is created. For more information, see Create buckets.

Background information

OSS provides cross-origin resource sharing (CORS) over HTML5 to implement cross-origin access. When OSS receives a cross-origin request (or an OPTIONS request) for a bucket, OSS reads the CORS rules of the bucket and checks the relevant permissions. OSS matches the rules one after another. If OSS finds the first match, OSS returns corresponding headers. If the request fails to match the CORS rules, OSS does not include CORS headers in the response.

Procedure

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the Basic Settings tab. In the Cross-Origin Resource Sharing (CORS) section, click Configure.
- 4. On the page that appears, click Create Rule. Then, in the Create Rule panel, configure the parameters in the following table.

Parameter	Required	Description
Sources	Yes	 The sources from which you want to allow cross-origin requests. When you configure the sources, take note of the following rules: You can configure multiple rules for sources. Separate multiple rules with line feeds. The domain names must include the protocol name, such as HTTP or HTTPS. You can use an asterisk (*) as the wildcard character. Each source can contain up to one asterisk (*). If a domain name does not use the default port, the domain name must contain the port number. Example: https://www.example.com:8080. The following examples show how to configure domain names: To match a specified domain name, enter the full domain name. Example: https://www.example.com. Use an asterisk (*) as a wildcard in the domain name to match second-level domains. Example: https://*.example.com. To match all domain names, enter only an asterisk (*) as the wildcard character.
Allowed Methods	Yes	The methods that cross-origin requests are allowed to use.
Allowed Headers	No	 The response headers for the allowed cross-origin requests. When you configure the headers, take note of the following rules: This parameter is in the key:value format and is not case-sensitive. Example: content-type:text/plain. You can configure multiple response headers. Separate multiple response headers with line feeds. Each rule can contain up to one asterisk (*) as the wildcard character. Set this parameter to an asterisk (*) if you do not have special requirements.
Exposed Headers	No	The response headers for allowed access requests from applications, such as an XMLHttpRequest object in JavaScript. Exposed headers cannot contain asterisks (*).
Cache Timeout (Seconds)	No	The timeout period of the response that is cached by the browser for an OPTIONS preflight request destined for specific resources. Unit: seconds.

Note You can configure up to 10 CORS rules for each bucket.

5. Click Confirm.

2.4.7. Configure lifecycle rules

You can configure lifecycle rules to regularly delete expired objects and parts to save storage costs.

Prerequisites

A bucket is created. For more information, see Create buckets.

Background information

Take note of the following items when you configure lifecycle rules for a bucket:

- After a lifecycle rule is configured, it is loaded within 24 hours and takes effect within 24 hours after it is loaded. Check the configurations of a rule before you save the rule.
- Objects that are deleted based on lifecycle rules cannot be recovered. Configure lifecycle rules based on your requirements.
- You can configure up to 100 lifecycle rules for a bucket in the Object Storage Service (OSS) console. To configure more rules, use OSS SDKs.

Procedure

1. Log on to the OSS console.

- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the Basic Settings tab. Find the Lifecycle section. Click Configure.
- On the page that appears, click Create Rule. In the Create Rule panel, configure the parameters. The following table describes the parameters.
 Parameters for unversioned buckets

Section	Parameter	Description
Basic Settings	Status	Specify the status of the lifecycle rule. Valid values: Enabled and Disabled.
	Applied To	Specify the objects to which the lifecycle rule applies. Valid values: Files with Specified Prefix and Whole Bucket . ③ Note If you select Files with Specified Prefix, you can configure multiple lifecycle rules for objects whose names contain different prefixes. If you select Whole Bucket, you can configure only one lifecycle rule for the bucket.
	Prefix	If you set Applied To to Files with Specified Prefix , you must specify the prefix of the objects to which the rule applies. For example, if you want that the rule applies to objects whose names start with img, enter <u>img</u> .
	Tagging	Specify tags. The rule applies only to objects that have the specified tags. For example, if you select Files with Specified Prefix and set Prefix to img, Key to a, and Value to 1, the rule applies to all objects that have the img prefix in their names and have the tag a=1.
Clear Policy	File Lifecycle	Configure rules for objects to specify when the objects expire. Valid values. Validity Period (Days) , Expiration Date , and Disabled . If you select Disabled , the configurations of File Lifecycle do not take effect.
	Delete	Specify the time when objects expire based on Validity Period (Days) or Expiration Date that you set for File Lifecycle. Expired objects are deleted.
Delete Parts	Part Lifecycle	Specify the operations that you want to perform on expired parts. You can set Part Lifecycle to Validity Period (Days), Expiration Date, or Disabled. If you select Disabled, the configurations of Part Lifecycle do not take effect. Important You must configure at least one of File Lifecycle and Part Lifecycle. If you select Tagging, Part Lifecycle is unavailable.
	Delete Parts	Specify the time when parts that match the rule expire based on Validity Period (Days) or Expiration Data that you set for Part Lifecycle . Expired parts are deleted.

· Parameters for versioned buckets

Configure the parameters in the **Basic Settings** and **Delete Parts** sections in the same way as the parameters configured for unversioned buckets. The following table describes only the parameters that are different from the parameters that you configure for unversioned buckets.

Section	Parameter	Description
Current Version	Clean Up Delete Marker	If you enable versioning for the bucket, you can configure the Clean Up Delete Marker parameter. Other parameters are the same as those you can configure for unversioned buckets. If you select Clean Up Delete Marker, and an object has only one version which is a delete marker, OSS considers the delete marker expired and removes the delete marker. If an object has multiple versions and the current version of the object is a delete marker, OSS retains the delete marker.
Previous Versions	File Lifecycle	Specify when previous versions expire. Valid values: Validity Period (Days) and Disabled. If you select Disabled, the configurations of File Lifecycle do not take effect.
	Delete	 Specify the number of days that objects can be retained as previous versions. The previous versions are deleted one day after they expire. Important If the current version of an object is deleted based on a lifecycle rule, OSS does not delete the current version but converts the current version to a previous version and adds a delete marker to the object. The delete marker becomes the current version of the object. If a previous version of an object is deleted based on a lifecycle rule, OSS deletes the previous version. If you configure a lifecycle rule to delete previous versions, delete markers that are stored as previous versions are also deleted.

5. Click Confirm.

2.4.8. Configure storage quota

If the capacity of a bucket reaches the specified storage quota, write operations such as PutObject, MultipartUpload, CopyObject, PostObject, and AppendObject cannot be performed on the bucket. This topic describes how to configure the storage quota of a bucket in Object Storage Service (OSS).

Prerequisites

A bucket is created. For more information, see Create buckets.

Background information

Take note of the following items when you configure the storage quota of a bucket:

- Before you configure the storage quota of a bucket, make sure that the quota does not limit your business because write operations cannot be
 performed if the bucket capacity reaches the quota.
- In general, it takes about an hour for OSS to determine whether the bucket capacity exceeds the storage quota. In some cases, it can take longer than one hour.

Procedure

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket that you want to manage.
- 3. Click the Basic Settings tab, find the Storage Quota section.

(?) Note You can view the storage guota of the bucket in this section.

- 4. Click **OK**.
- 5. On the Modify OSS Bucket page, modify the storage quota of the bucket.
- Units: TB or GB.
- Valid values: 1 to 2000000 TB or 1 to 2048000000 GB
- 6. Click Submit
- After you submit, you can click **Back to Console** in the pop-up dialog box to go back to the **Overview** page.
- 7. View the storage quota of a bucket
 - i. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the bucket whose quota you want to view.
- ii. On the Details tab, find the Basic Settings section.
- iii. View the storage quota of the bucket.

(2) Note A message that indicates the storage quota of the bucket is displayed below **Basic Settings**. Example: The storage quota of the bucket is 2 (TB). Please ensure that the quota is larger than the storage capacity that you use.

2.4.9. Configure bucket tagging

Bucket tags can be configured to classify and manage buckets. For example, you can list buckets that have specific tags or configure the access control list (ACL) of buckets that have specific tags.

Background information

The bucket tagging feature uses a key-value pair to identify a bucket. You can add tags to buckets that are used for different purposes and manage the buckets by tags.

- Only the bucket owner or authorized Resource Access Management (RAM) users can configure tagging for the bucket. Otherwise, 403 Forbidden is
 returned with the error code AccessDenied.
- A tag is a key-value pair. You can configure up to 20 tags for a bucket.
- Each tag must have a key. The key of a tag can be up to 64 bytes in length and cannot start with http://, or http://, or http://.
- The value of a tag can be up to 128 bytes in length and can be empty.
- The key and value of a tag must be UTF-8-encoded.

Procedure

- 1. View publishing history.
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure tagging.
- 3. Click the **Basic Settings** tab. Find the **Bucket Tagging** section.
- 4. Click **OK**.
- 5. Add tags to the bucket based on the naming conventions. You can click the + icon to add multiple tags to a bucket.
- 6. Click Save.

2.4.10. Configure zone-disaster recovery

Zone-disaster recovery synchronizes objects in a bucket to a secondary bucket with the same name as the bucket in the secondary cluster to enable cluster-level disaster recovery. If zone-disaster recovery is enabled, a secondary bucket with the same name as the primary bucket is automatically created and objects in the primary bucket are automatically synchronized to the secondary bucket.

Prerequisites

A bucket is created. For more information, see Create buckets.

Background information

By default, objects that are encrypted by using CMKs managed by Key Management Service (KMS) in the source bucket cannot be synchronized to the destination bucket, even if zone-disaster recovery is enabled.

To synchronize objects encrypted by using CMKs managed by KMS to the destination bucket, you must create a RAM role that has permissions to manage and decrypt the objects by using CMKs and attach the role to your account or RAM user. For more information, seeGrants permissions to a role.

- 1. Log on to the OSS console. For more information, see Log on to the OSS console.
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the Basic Settings tab and go to the Cluster-disaster Recovery section.

4. Click Configure to configure zone-disaster recovery.

• To synchronize objects encrypted by using CMKs managed by KMS to the destination bucket, turn on Cluster-disaster Recovery, select the KMS-based Encryption check box, and then specify a CMK ID and a RAM role. The following table describes the CMK ID and RAM Role Name parameters.

Parameter	Description
CMK ID	Select a KMS-managed CMK for object encryption and decryption. You can specify this parameter only when the KMS-based Encryption check box is selected. Make sure that you have created a KMS CMK in the KMS console.
	Attach a role to the OSS service. The linked role must have permissions to perform operations on encrypted objects and encrypt and decrypt objects by using KMS-managed CMKs. You can specify this parameter only when the KMS-based Encryption check box is selected. To attach a role that has the required permissions, you can use the predefined role AliyunOSSPrivateCloudDrsSyncRole or create a custom role. For more information about how to grant permissions to a role, see Grants permissions to a role.
RAM Role Name	 Use the predefined RAM role
	OSS provides the predefined role AliyunOSSPrivateCloudDrsSyncRole. You can use this predefined role when you configure a RAM role during bucket creation.
	Create a custom role
	You can click Create Role to create a custom role.

• To disable zone-disaster recovery, turn off Cluster-disaster Recovery.

5. Click Save.

2.4.11. Versioning

OSS allows you to configure versioning for a bucket to protect objects that are stored in the bucket. After you enable versioning for a bucket, data that is overwritten or deleted in the bucket is saved as a previous version. After you configure versioning for a bucket, you can recover objects in the bucket to a previous version to protect your data from being accidentally overwritten or deleted.

Background information

When versioning is enabled, OSS specifies a unique ID for each version of all objects in a bucket. You can also download a previous version of an object or recover the previous version as the current version at any time based on the version ID.

Enable versioning

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the bucket that you want to manage.
- 3. Click the Basic Settings tab. In the Back-to-Origin section, click Configure.
- 4. Click Enable, and then click Save

If you no longer need more object versions in a versioning-enabled bucket, you can select Suspend. In this case, OSS generates the ? Note version ID of null for new versions and does not generate previous versions. Existing previous versions are retained.

Recover previous versions

- You can recover an object to a specified previous version.
- 1. Log on to the OSS console
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket in which the object you want to recover is stored. 3. Click Files. In the upper-right corner, set Display Previous Versions to Show.
- 4. Click Recover in the Actions column corresponding to the previous version to which you want to recover. OSS copies the specified version of the object to the current directory. The existing current version is overwritten. The specified previous version becomes the current version.

Download previous versions

- You can download a specified previous version of an object.
- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the bucket in which the version you want to download is stored.
- 3. Click Files. In the upper-right corner, set Display Previous Versions to Show.
- 4. Click the object version you want to download. In the panel that appears, click Download on the right side of Signed URL.
- 5. Select the location where you want to store the downloaded version and then click Save.

Delete previous versions

- We recommend that you delete unnecessary previous versions in a timely manner to minimize storage costs.
- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket that you want to manage.
- 3. Click Files. In the upper-right corner, set Display Previous Versions to Show
- 4. Choose More > Permanently Delete in the Actions column corresponding to the previous version that you want to delete. In the message that appears, click **OK**.

To delete multiple previous versions of an object, select the previous versions that you want to delete and choose Batch Operation > Permanently Delete

5. Click OK.

() Important

- Deleted previous versions cannot be recovered. Exercise caution when you perform this operation.
- If you delete the current version of an object, the latest previous version becomes the current version.
- You can also configure lifecycle rules to automatically delete previous versions on a regular basis. For more information, see Configure lifecycle rules.

2.4.12. Configure server-side encryption

Object Storage Service (OSS) supports server-side encryption. When you upload an object to a bucket for which server-side encryption is enabled, OSS encrypts and then stores the object. When you download an encrypted object, OSS decrypts and then returns the decrypted object. A header is added to the response to indicate that the object is encrypted on the OSS server.

Background information

- OSS supports the following encryption methods:
- Server-side encryption by using SSE-KMS
- OSS uses the default customer master key (CMK) managed by KMS or a specified CMK to encrypt objects. The CMK is managed by KMS to ensure confidentiality, integrity, and availability at minimal costs.
- Server-side encryption that uses OSS-managed keys (SSE-OSS)
- OSS uses data keys to encrypt objects and manages the data keys. In addition, OSS uses master keys that are regularly rotated to encrypt data keys. You can enable server-side encryption in the OSS console by using one of the following methods:
- Method 1: Enable server-side encryption when you create a bucket
- Method 2: Enable server-side encryption on the Basic Settings tab

Method 1: Enable server-side encryption when you create a bucket

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click Create Bucket.
- 3. In the **Create Bucket** panel, configure the parameters.
 - You can set the following parameters to configure server-side encryption for the bucket.
 - Server-Side Encryption: Specify the encryption methods.
 - None: Server-side encryption is not performed.
 - OSS-Managed: Keys managed by OSS are used to encrypt objects in the bucket.
 - KMS: CMKs managed by KMS are used to encrypt objects in the bucket.
 - Encryption Algorithm: Select an encryption algorithm
 - CMK: You can set this parameter if Encryption Method is set to KMS. OSS uses the specified CMK to encrypt objects in the bucket.
 - (?) Note To select a CMK ID for server-side encryption, you must create the CMK in the KMS console.

4. Click Confirm.

Method 2: Enable server-side encryption on the Basic Settings tab

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the bucket that you want to manage.
- 3. Click the Basic Settings tab. Find the Server-side Encryption section.
- Click Configure and set the following parameters:
 - Encryption Method: Specify the encryption method.
 - **None**: Server-side encryption is not performed.
 - OSS-Managed: Keys managed by OSS are used to encrypt objects in the bucket.
 - KMS: CMKs managed by KMS are used to encrypt objects in the bucket.
 - Encryption Algorithm: Select an encryption algorithm.
 - CMK: You can set this parameter if Encryption Method is set to KMS. OSS uses the specified CMK to encrypt objects in the bucket.

(?) Note To select a CMK ID for server-side encryption, you must create the CMK in the KMS console.

5. Click Save.

() Important The configurations of the default encryption method for a bucket do not affect the encryption configurations of existing objects within the bucket.

2.4.13. Bind a VPC

You can bind your bucket to a specified virtual private cloud (VPC) network to allow only requests from IP addresses within the VPC network to access your bucket.

Prerequisites

A VPC network is created. For more information, see the "Create a VPC" topic in Apsara Stack VPC User Guide.

Procedure

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the bucket that you want to manage.
- 3. Click the Overview tab. Click Bind VPC in the VPC Info section.
- 4. On the **Bind VPC** page, select the VPC network that you create.
- You can also click **Create VPC** to create a new VPC network.

5. Click Submit.

After you bind a VPC, you can click **Back to Console** in the pop-up dialog box to go back to the **Overview** page.

2.4.14. Configure CRR rules

Cross-region replication (CRR) allows you to automatically and asynchronously (in near real-time) replicate Object Storage Service (OSS) objects from a bucket in a region to another bucket in another region. CRR also allows you to replicate operations, such as the creation, overwriting, and deletion of objects, from a source bucket to a destination bucket.

Prerequisites

The source bucket and destination bucket are created. For more information, see Create a bucket.

Background information

CRR can help you meet compliance requirements for cross-region disaster recovery or data replication. Objects in the destination bucket are exact replicas of objects in the source bucket. The objects have the same object names, object content, and object metadata, such as the creation time, owner, user metadata, and access control lists (ACLs).

Procedure

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the bucket for which you want to configure a CRR rule.
- 3. On the bucket details page, click the Basic Settings tab. Find the Cross-Region Replication section.
- 4. Click Enable. In the Cross-Region Replication panel, configure the parameters described in the following table.

Parameter	Description
Source Region	The region in which the current bucket is located.
Source Bucket	The name of the current bucket.
Destination region	The region in which the destination bucket is located. The source and destination buckets for CRR must be located in different regions. You cannot configure a CRR rule to replicate data between buckets that are located in the same region.
Destination Bucket	The destination bucket to which you want to replicate data. The source and destination buckets specified in a CRR rule are not allowed to be specified in other CRR rules. For example, if you configure a CRR rule to replicate data from Bucket A to Bucket B, Bucket A and Bucket B are not allowed to be configured in other CRR rules.
Applied To	 The objects that you want to replicate to the destination bucket. Valid values: All Files in Source Bucket: All objects in the source bucket are replicated to the destination bucket. Files with Specified Prefix: Only objects whose names contain one of the specified prefixes are replicated to the destination bucket. For example, if you have a subdirectory named management/ in the root directory of the source bucket and want to replicate objects in a subdirectory named abc/ in management/, you can set the prefix to management/abc/. You can specify up to 10 prefixes.
Operations	 The operations that you want to replicate. Valid values: Add/Change: OSS replicates only object creation and update operations from the source bucket to the destination bucket. Add/Delete/Change: OSS replicates object creation, update, and deletion operations from the source bucket to the destination bucket.
Replicate Historical Data	Specifies whether to replicate historical data (data that exists in the source bucket before you enable CRR) to the destination bucket. • Yes: OSS does not replicate historical data to the destination bucket. • Important When historical data is replicated, objects that are replicated from the source bucket may overwrite objects that have the same names in the destination bucket. To prevent data loss, we recommend that you enable versioning for the source and destination buckets. • No: OSS replicates only objects that are uploaded or updated after the CRR rule takes effect to the destination bucket.

5. Click Confirm.

- ? Note
 - After you configure a CRR rule, the replication task starts in 3 to 5 minutes. Then, you can view the replication progress.
 - In CRR, data is asynchronously replicated in near real time. The period of time that is required to replicate data from the source bucket to the destination bucket may range from a few minutes to a few hours, depending on the data size.
 - If you want to replicate an object encrypted based on KMS from the source bucket to the destination bucket, KMS must be configured for the destination bucket. Otherwise, data replication fails.

2.4.15. Configure cross-cloud replication

You can use the cross-cloud replication feature to synchronize OSS data between two clouds. This topic describes how to configure cross-cloud replication.

Step 1: Obtain the parameters of the destination cloud

Before you configure cross-cloud replication, you must obtain the required parameters of the destination cloud.

- 1. Log on to the Apsara Uni-manager Operations Console of the destination cloud.
- For more information about how to log on to the Apsara Uni-manager Operations console, see Log on to the Apsara Uni-manager Operations

console in Operations and Maintenance Guide

- 2. In the left-side navigation pane, choose **Product Management > Products**.
- 3. Click OSS O&M.
- 4. In the left-side navigation pane, choose Service O&M OSS > Synchronization Management > Cross-Cloud Synchronization.
- In the upper-right corner, select the destination cluster from the Cluster drop-down list, and then click View Parameters of the Current Cloud. Record the information displayed in the Parameters of the Current Cloud dialog box.

Step 2: Configure cross-cloud synchronization for the source cloud.

After you obtain the required parameters of the destination cloud, you must configure cross-cloud synchronization for the source cloud in the Apsara Uni-manage Operations Console.

- Log on to the Apsara Uni-manager Operations Console of the destination cloud. For more information about how to log on to the Apsara Uni-manager Operations console, see Log on to the Apsara Uni-manager Operations console in Operations and Maintenance Guide.
- In the left-side navigation pane, choose Product Management > Products.
- 3. Click OSS O&M.
- 4. In the left-side navigation pane, choose Service O&M OSS > Synchronization Management > Cross-Cloud Synchronization.
- In the upper-right corner, click Create. In the Create Cross-Cloud Synchronization Task dialog box, add the obtained parameters of the destination cloud.
- 6. Click Submit.

Wait a few minutes until the cross-synchronization configurations take effect.

Step 3: Configure cross-cloud replication in the Apsara Uni-manager Management Console of the source cloud

After the cross-cloud synchronization configurations take effect, you must configure cross-cloud replication in the Apsara Uni-manager Management Console.

1. Log on to the OSS console.

- In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure cross-cloud replication.
- 3. On the bucket details page, click the Basic Settings tab. In the Cross-Cloud Replication section, click Configure.
- 4. In the Cross-Cloud Replication panel, configure the parameters described in the following table.

Parameter	Description
Source Region	The region in which the source bucket is located.
Source Bucket	The name of the source bucket.
Destination Cloud	Enter the name of the destination cloud obtained in Step 1.
Destination Cloud Address	Enter the value of Location of the destination cloud obtained in Step 1.
Destination Bucket	Enter the name of the destination bucket.
Applied To	 Select the objects that you want to synchronize. All Files in Source Bucket: All objects within the source bucket are synchronized to the destination bucket. Files with Specified Prefix: Only objects whose names contain one of the specified prefixes are synchronized to the destination bucket. Click Add. You can add up to 10 prefixes.
Operations	 Select a synchronization policy. Add/Change: Only newly added and changed data is synchronized from the source bucket to the destination bucket. Add/Delete/Change: All changes to data including creation, modification, and deletion of objects are synchronized from the source bucket to the destination bucket.
Replicate Historical Data	 Specify whether to synchronize historical data that is generated before you enable cross-cloud replication. Yes: OSS synchronizes historical data to the destination bucket. No: Only objects that are uploaded or updated after cross-cloud replication is enabled are synchronized to the destination bucket.

5. Click **OK** to save your settings.

2.4.16. Configure retention policies

Object Storage Service (OSS) allows you to configure a write once read many (WORM) policy for a bucket. During the retention period specified in the policy, you can upload objects to the bucket and access the objects in the bucket but cannot delete or modify the objects in the bucket.

- 1. Log on to the OSS console
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure a retention policy.
- 3. Click the **Basic Settings** tab. In the **Retention Policy** section, click **Configure**.
- 4. Click Create Policy
 - Only time-based policies are supported.
- 5. Configure the **Retention Period** parameter and click **OK**.

? Note

The new retention period must not be shorter than the original retention period. The retention period ranges from 1 to 25,550 days

2.4.17. Log management

2.4.17.1. Configure logging

When you access Object Storage Service (OSS), a large number of access logs are generated. You can use the logging feature to store OSS access logs in a specified bucket.

Prerequisites

A bucket is created. For more information, see Create buckets.

Procedure

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the Basic Settings tab. Find the Logging section.
- 4. Click Configure. Turn on the Logging switch. Select Destination Bucket and configure Log Prefix.
- Destination Bucket: Select the bucket in which you want to store access logs from the drop-down list. You must be the owner of the selected bucket, and the selected bucket must be in the same region as the bucket for which logging is enabled.
- Log Prefix: Enter the prefix and directory where the access logs are stored. If you specify log/<targetPrefix as the prefix, access logs are stored in the log/ directory.

5. Click Save.

2.4.17.2. Real-time log query

A large number of logs are generated when Object Storage Service (OSS) resources are accessed. OSS uses Log Service to help you query and collect statistics for OSS access logs and audit access to OSS in the OSS console, track exception events, and troubleshoot problems. This helps you improve work efficiency and make informed decisions.

Advantages

- Pushes logs to Log Service within three minutes and allows you to view real-time logs in the OSS console.
- Provides log analysis and common analysis reports so that you can easily query data.
- Allows you to query and analyze raw logs in real time and filter logs by bucket name, object name, API operation, or time.

Prerequisites

Log Service is activated and is authorized to access OSS.

Enable real-time log query

- () Important
 - After you enable real-time log query for a bucket for the first time, you must wait for about one minute and then refresh the page to use this feature.
 - When you enable real-time log query, the system creates Log Service projects.
- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name to which you want to upload objects.
- 3. Click Logging and then click Real-time Log Query. On the Real-time Log Query tab, click Activate Now.

Specify the retention period of logs

By default, logs are retained for seven days. You can modify the retention period based on your business requirements.

- 1. Log on to the OSS console
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name to which you want to upload objects.
- 3. Click Logging and then click Real-time Log Query. On the Real-time Log Query tab, click Config Log Retention Time.
- In the Config Log Retention Time dialog box, modify the retention time. Then, click OK. Data can be retained for 7 to 3,000 days.

Description

Query real-time logs

OSS uses Log Service to help you query and collect statistics for OSS access logs and audit access to OSS in the OSS console, track exception events, and troubleshoot problems. This helps you improve work efficiency and make informed decisions.

- Method 1: Query real-time logs on the Original Log tab
- i. Log on to the OSS console.
- ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the bucket that you want to manage.
- iii. Click Logging and then click Real-time Log Query. On the Real-time Log Query tab, click the Original Log tab and enter the query condition.
 - A query condition consists of a query statement and an analytic statement in the query statement | analytic statement format. Example: * | SELECT status, count (*) AS PV GROUP BY status . The query statement and the analytic statement are separated by a vertical bar (|). The query statement uses proprietary syntax of Log Service.

A query statement can be individually executed. However, an analytic statement must be executed together with a query statement. In other words, analysis is performed based on the query results or the complete data. For more information, see the "Search and Analysis section of the Log Service" section in **CDS User Guide**.

Statement

Query statement	A query statement specifies one or more query conditions, and then returns the logs that meet the specified conditions. A query statement can be a keyword, a numeric value, a numeric value range, a space character, or an asterisk (*). If you specify a space character or an asterisk (*) as the query statement, no conditions are specified and all logs are returned.
Analytic statement	An analytic statement is used to aggregate or analyze all log data or the log data that meets the specified query conditions in a Logstore.

iv. Click 15 Minutes(Relative) to specify the time range for the query statement.

You can select a relative time, a time frame, or a custom time range. However, the time range that you can specify is only accurate to the minute at most. If you want to use a time range that is accurate to the second, you must specify the time range in the analytic statement. Example: * | SELECT * FROM log WHERE __time_>1558013658 AND __time_< 1558013660 .

v. Click Search & Analyze.

Query results contain query and analysis results in a log distribution histogram, on the Raw Logs tab, and on the Graph tab. You can also perform operations on the results. For example, you can configure alerts and create saved searches.

- Log distribution histogram
 - The log distribution histogram shows the distribution of returned logs in different periods of time.
- Alert

You can click **Save as Alert** to configure alerts for query and analysis results.

Saved search

You can click Save Search to save a query statement as a saved search.

Raw log

You can click the Table or Raw Data option on the **Raw Logs** tab to analyze the distribution of a field over a period of time, view the context of the specified log in the raw file, monitor the log content in real time, and extract key log information.

Graph

On the **Graph** tab, you can view the visual query and analysis results, add charts to the dashboard, download logs, and configure interactive behaviors.

LogReduce tab

On the LogReduce tab, you can click Enable LogReduce to cluster similar logs during log collection.

- Method 2: Query real-time logs on the Dashboard tab
- i. Log on to the OSS console
- ii. In the left-side navigation pane, click Buckets. On the Buckets page, click the name of the bucket that you want to manage.
- iii. Click the **Dashboard** tab to analyze logs.

Dashboard allows you to view four reports that are immediately available.

- Access Center: displays the overall operating status including the PV, UV, traffic, and distribution of access over the Internet.
- Audit Center: displays statistics of object operations including read, write, and delete operations.
- Operation Center: displays statistics of access logs including the number of requests and distribution of failed operations.
- Performance Center: displays statistics of performance including the performance of downloads and uploads over the Internet, the
 performance of transmission over different networks or with different object sizes, and the list of differences between stored and downloaded
 objects.

Disable real-time log query

() Important If you disable real-time log query, the system does not delete Log Service projects. Delete these projects before you disable real-time log query.

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name to which you want to upload objects.
- 3. Click Logging and then click Real-time Log Query. On the Real-time Log Query tab, click Disable Real-time Log Query.

2.4.18. Image processing

2.4.18.1. Configure image styles

You can encapsulate multiple image processing (IMG) parameters in a style and perform complex IMG operations by using the style.

Background information

Up to 50 styles can be created for a bucket. These styles can be used only for image objects in the bucket.

Create an image style

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the Image Processing (IMG) tab. On the page that appears, click Create Rule.
- 4. In the Create Rule panel, configure the style.
- You can create an image style by using one of two methods: Basic Edit or Advanced Edit
- Basic Edit: provides the graphical user interface (GUI) to choose the IMG methods such as resizing an image, adding a watermark, and modifying the image format.
- Advanced Edit: uses the API code to edit the IMG features that you want to use to process images. Format: image/action1,parame_value1/action2,parame_value2/...

Example: image/resize, p_63/quality, q_90 indicates that the image is scaled down to 63% of the source image, and then the relative quality of the image is set to 90%.

Once If you want to add image and text watermarks to images at the same time by using a style, use Advanced Edit to create the style.

5. Click Confirm.

Apply styles

After a style is created, you can use the style to process your image objects in the bucket.

- 1. On the Overview page, click the Files tab
- 2. Click the name of the image that you want to process.
- 3. In the View Details panel, select an image style from the Image Style drop-down list.

You can view the processed image in the View Details panel. Right-click the image and click Save As to save the image to your local disk.

Simplify IMG URLs that include style parameters

An IMG URL that has a style contains a file access URL, style parameters, and a style name. Example: https://image-demo.oss-cn-qd-ase-d01-a.mytestinc.com/example.jpg?x-oss-process=style/small . You can replace the ?x-oss-process=style/ field with a custom delimiter to simplify the IMG URL. For example, if you specify the delimiter as an exclamation point (!), the IMG URL can be simplified to https://image-demo.oss-cn-qd-ase-d01-a.mytestinc.com/example.jpg!small

1. In the Buckets page, click the Image Processing (IMG) tab.

- 2. Click Access Settings
- 3. On the Access Settings panel, select one of the Delimiters.
- Only hyphens (-), underscores (_), forward slashes (/), and exclamation points (?) can be used as delimiters
- 4. Click Confirm

2.4.18.2. Configure source image protection

OSS provides the source image protection feature to protect your images from being used by unauthorized anonymous requesters. After you enable source image protection for your bucket, anonymous requesters can access images in the bucket only by adding style parameters in the requests or by using signed URLs.

Background information

You can use the following methods to access the images:

- Use the file URL that contains the style parameters in the format of https://BucketName.Endpoint/ObjectName?x-oss-process=style/StyleName .
- Use the file URL that contains a signature in the format of https://BucketName.Endpoint/ObjectName?Signature .

Procedure

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the Image Processing (IMG) tab. On the page that appears, click Access Settings.
- 4. In the Access Settings panel, turn on Protect Source Image File and configure the parameters described in the following table.

Parameter	Description
Protected File Extensions	Select a file suffix from the Protected File Extensions drop-down list. All objects in the bucket that match the specified suffix are protected.
Delimiters	If you set the custom delimiter, use the delimiter to replace ? x-oss-process=style/ to further simplify the IMG URL. OSS supports the following delimiters: hyphens (-), underscores (_), forward slashes (/), and exclamation points (!). Click the check box before the delimiter that you want to select. For example, if you set the delimiter to an exclamation point (!), the IMG URL can be simplified to <pre>http(s)//:BucketName.Endpoint/ObjectName!StyleName</pre> .

5. Click Confirm

FAO

• Q: Why is HTTP status code 403 returned when I directly access a protected image, whereas HTTP status code 200 is returned when I access the image over Content Delivery Network (CDN)?

A: One possible cause is that the request is redirected to access a private bucket over CDN. Source image protection is applicable only to objects that are accessed by anonymous users

• Q: How can my source image still be accessed by using a signed URL when source image protection is enabled for the image?

Source image protection applies only to objects that can be accessed by anonymous users. Access by signed URLs is not anonymous. Therefore, the source image can be accessed by using a signed URL even if you enable source image protection.

2.4.19. Grants permissions to a role

In cluster-disaster recovery scenarios, if you want to synchronize objects encrypted by using CMKs managed by KMS to the destination bucket, you must use a role that has permissions to manage and decrypt the objects by using CMKs. In this case, you can use the predefined role AliyunOSSPrivateCloudDrsSyncRole or create a custom role.

Create a RAM role

You can create a service-linked role that is attached to the predefined role AliyunOSSPrivateCloudDrsSyncRole to authorize OSS to view and perform operations on other resources in the organization.

- 1. Go to the Service-linked Roles page.
- i. Log on to the Apsara Uni-manager Management Console.d
- ii. In the top navigation bar, click Configurations.
- iii. In the left-side navigation pane, click Service-linked Roles.
- 2. On the Service-linked Roles page, click Create Service-linked Role.
- On the page that appears, select an organization and set Service Name to OSS. The organization of the service-linked role must be the same as that of the bucket in which the objects need to be encrypted by using CMKs are stored.
- 4 Click OK

On the Service-linked Roles page, you can click Details in the Actions column corresponding to the predefined role AliyunOSSPrivateCloudDrsSyncRole to view the information about the role.

· Click the Role Details tab to view the following information:

- Role Name: AliyunOSSPrivateCloudDrsSyncRole
- Trust Policy:

```
{
   "Statement": [
   {
        "Action": "sts:AssumeRole",
        "Effect": "Allow",
        "Principal": {
            "Service": [
            "oss.aliyuncs.com"
        ]
      }
   }
   }
},
"Version": "1"
```

- Click the Role Policy tab and view the following information:
 - Policy Name: AliyunOSSPrivateCloudDrsSyncRolePolicy
 - Description:

}

```
"Version": "1",
"Statement": [
 {
      "Action": "oss:*",
      "Effect": "Allow",
      "Resource": "*"
  },
  {
      "Action": "oss:PutBucket*",
      "Effect": "Deny",
      "Resource": "*"
  },
      "Action": [
           "kms:Encrypt",
           "kms:Decrypt",
           "kms:DescribeKey",
      "kms:GenerateDataKey"],
"Resource": "*",
       "Effect": "Allow"
  }]
```

Create a custom role

You can create custom roles to control the permissions of users at different granularities. To synchronize objects encrypted by using CMKs managed by KMS to the destination bucket, the custom role that you create must have permissions to manage, encrypt, and decrypt the objects by using CMKs. You can perform the following steps to create the custom role:

- 1. Go to the Role authorization page.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, click Enterprise.
- iii. In the left-side navigation pane, choose **Permissions > Role Permissions**.
- 2. In the upper-left corner of the Role authorization page, click Create Custom Role.
- 3. On the Create Custom Role page, click Advanced Settings. In the Advanced Settings section, click Create RAM Role & Attach Policy.
- 4. On the Create RAM Role & Attach Policy page, set the parameters described in the following table and then click Next.

Parameter	Description
Role name	The name of the role that you want to create. The name can be up to 64 characters in length and can contain only letters and digits.
Description	Optional. The description of the role. The description can be up to 500 characters in length and can contain letters, digits, commas (,), semicolons (;), and underscores (_).
Sharing Scope	The scope in which the role is visible. Valid values: Global: This role is visible and valid for all organizations.

5. In the Configure Custom Policy step, select an existing policy or create a new policy, and then click **Next**.

i. On the Policies page, click Create Policy

ii. In the Create Policy dialog box, configure the parameters described in the following table.

Parameter	Description
Policy Name	The name of the policy you want to create. The name must be 2 to 50 characters in length and can contain only letters and digits. The name must be unique in the system.
Description	The description of the policy.
Sharing Scope	The scope in which the policy is visible. Valid values: Global : This role is visible and valid for all organizations.
Policy Content	<pre>The permissions defined by this policy. The policy content is in JSON format. The following example shows how to specify the content of a policy: { "Version": "1", "Statement": [{ "Action": "oss:*", "Effect": "Allow", "Resource": "*" }, { "Action": "oss:PutBucket*", "Effect": "Deny", "Resource": "*" }, { "Action": ["kms:Decrypt", "kms:Decrypt", "kms:DecribeKey", "kms:GenerateDataKey"], "Resource": "*", "Effect": "Allow" "J] } </pre>

iii. Click OK.

- 6. Modify the trust policy and then click Next.
- i. Click Modify Trust Policy.
- ii. The trust policy is in JSON format. The following example shows how to modify the trust policy:

```
"Statement": [
  {
   "Action": "sts:AssumeRole",
   "Effect": "Allow",
    "Principal": {
     "Service": [
       "oss.aliyuncs.com"
     1
 }
],
"Version": "1"
```

iii. Click Save Changes.

Confirm the information about the custom role to be created, and then click Create. After the custom role is created, you can view the information about the role on the Role authorization page.

If you no longer use the role that you create, click Delete in the Actions column corresponding to the role in the Role authorization page to delete the role.

- 8. Create a RAM role for the custom role.
 - i. On the Role authorization page, click the name of the custom role that you create.
 - ii. On the details page of the custom role, click the RAM Role tab and then click Create.
- iii. In the Create dialog box, select an organization.
 - In scenarios where you use the custom role to synchronize objects encrypted by using CMKs in a bucket, the organization of the RAM role must be the same as that of the bucket.

iv. Click OK.

After the RAM user is created, you can view the information about the RAM user on the RAM Role tab.

If you no longer use the RAM role that you create, click **Remove** in the **Actions** column corresponding to the RAM role on the **RAM Role** page to delete the RAM role.

2.5. Objects

2.5.1. Search for objects

You can search for objects whose names contain specific prefixes in buckets or folders in the OSS console.

Prerequisites

Objects are uploaded to the bucket. For more information, see Upload objects.

Background information

When you search for objects based on a prefix, search strings are case-sensitive and cannot contain forward slashes (/). You can search for objects only in the root directory of the current bucket or in the current directory. Directories and objects stored in subdirectories cannot be searched

Procedure

1. Log on to the OSS console

- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name to which you want to upload objects.
- 3 Click the Files tab
- 4. Search for objects.
 - Search for objects and subdirectories in the root directory.

In the upper-right corner, enter the prefix to search in the search box and press Enter or click the 📿 icon to search for related objects. Objects and subdirectories whose names contain the specified prefix within the root directory of the bucket are displayed.

· Search for objects and subdirectories in a specified directory

Click the directory in which the objects or subdirectories that you want to search for are stored. In the upper-right corner, enter the prefix to search in the search box and press Enter or click the 🔘 icon to search for related objects. Objects and subdirectories whose names contain the specified prefix within the current directory are displayed.

2.5.2. Configure object ACLs

You can configure the ACL of an object in the OSS console to control access to the object.

Prerequisites

An object is uploaded to the bucket. For more information, see Upload objects.

Procedure

1. View publishing history.

- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the Files tab
- Click the name of the object whose ACL you want to configure. In the View Details panel, click Set ACL on the right side of File ACL. You can also choose More > Set ACL in the Actions column corresponding to the object whose ACL you want to configure.
- In the Set ACL panel, configure the ACL of the object. You can set the ACL of the object to one of the following values:
 - Inherited from Bucket: The ACL of the object is the same as that of the bucket.
 - Private: Only the owner or authorized users of this bucket can read and write objects in the bucket. Other users, including anonymous users cannot access the objects in the bucket without authorization.
 - Public Read: Only the owner or authorized users of this bucket can write objects in the bucket. Other users, including anonymous users can only read objects in the bucket.
 - Public Read/Write: All users, including anonymous users can read and write objects in the bucket. Fees incurred by such operations are charged to the owner of the bucket. Exercise caution when you set the object ACL to this value.

6. Click Confirm

2.5.3. Configure object metadata

Objects that are stored in Object Storage Service (OSS) consist of keys, data, and object metadata. Object metadata describes the object. Object metadata includes standard HTTP headers and user metadata. You can create custom HTTP request policies such as object cache policies and forced object download policies by configuring standard HTTP headers. You can also configure user metadata to identify the purposes or attributes of objects.

Background information

You can configure object metadata for up to 100 objects at a time in the OSS console.

Procedure

1. Log on to the OSS console.

- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the Files tab.
- 4. Use one of the following methods to open the Set HTTP Header panel:
 - Configure HTTP headers for multiple objects
 - Select one or more objects. Choose Batch Operation > Set HTTP Header.
 - Configure HTTP headers for a single object
 - Find the object for which you want to configure HTTP headers and choose More > Set HTTP Header in the Actions column.
- 5. In the Set HTTP Header panel, configure the parameters. The following table describes the parameters.

Parameter	Description
Content-Type	The type of the object. The browser determines the default method that is used to open an object based on the object type. For example, the value of the Content-Type parameter for a GIF image is <code>limage/gif</code> .

Content-Encoding	The encoding method of the object. You must specify this header based on the encoding type of the object. Otherwise, the browser that serves as the client may fail to parse the encoding type of the object, or the object may fail to be downloaded. If the object is not encoded, leave this parameter empty. Default value: identity. Valid values: identity. OSS does not compress or encode the object. gzip: OSS uses the LZ77 compression algorithm created by Lempel and Ziv in 1977 and 32-bit cyclic redundancy check (CRC) to encode the object. compress: OSS uses the LZPY compression algorithm to encode the object. deflate: OSS uses the Zlib library and the deflate algorithm to encode the object. br: OSS uses the Brotli algorithm to encode the object. For more information about Content-Encoding, see RFC 2616. Important If you want the static web page objects, such as HTML, JavaScript, XML, and JSON objects to be compressed into GZIP objects when you access these objects, you must leave this parameter empty and add the <u>Accept-Encoding</u> : gzip header to your request.
Content-Language	The language of the object content. For example, if the content of an object is written in simplified Chinese, you can set this parameter to $rh-CN$.
Content-Disposition	 The method used to access the object. Valid values: inline: The object is directly opened in the browser. To ensure that an image object or a web page object is previewed but not downloaded when the object is accessed, you must set Content-Disposition to inline and use the custom domain name mapped to the bucket to access the object. attachment: The object is downloaded to the local computer. For example, if this header is set to attachment; filename="example.jpg", the object is downloaded to the local computer. After the object is downloaded, the local file is named example.jpg . For more information about Content-Disposition, see RFC 2616.
Cache-Control	 The cache configurations for the object. Valid values: no-cache: The object can be cached on the client or on the browser of the proxy server. However, each time you access the object, OSS checks whether the cached object is available. If the cache is available, you can directly access the cache. Otherwise, the access request is sent to OSS. no-store: All content of the object is not cached. public: All content of the object is cached. private: All content of the object is cached only on the client. For more information about Cache-Control, see RFC 2616.
Expires	The absolute expiration time of the cache in Greenwich Mean Time (GMT). Example: 2022-10- 12T00:00:00.0002 . If max-age= <seconds> is set for Cache-Control, max-age=<seconds> takes precedence over Expires.</seconds></seconds>
User Metadata	Add the descriptive information for the object. You can add multiple user metadata headers for an object. However, the total size of user metadata cannot exceed 8 KB. When you add user metadata, user metadata headers must be prefixed with x-oss-meta- and assigned values. Example: x-oss-meta-last-modified:20200909u

6. Click Confirm.

2.5.4. Create directories

You can use the OSS console to create and simulate basic features of directories in Windows. This topic describes how to create a directory by using the OSS console

Prerequisites

A bucket is created. For more information, see Create buckets.

Background information

OSS does not use a hierarchical structure to store objects, but instead uses a flat structure. All elements are stored in buckets as objects. To help organize objects and simplify management, the OSS console displays objects whose names end with a forward slash (/) as directories. The objects can be uploaded and downloaded. You can use directories in the OSS console in the same manner as you use directories in Windows.

② Note The OSS console displays objects whose names end with a forward slash (/) as directories, regardless of whether these objects contain data. The objects can only be downloaded by calling an API operation or by using OSS SDKs

Procedure

- 1. Log on to the OSS console.
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the Files tab. On the page that appears, click Create Folder.
- In the Create Folder panel, enter the directory name. The directory name must comply with the following conventions:

 - The name can contain only UTF-8 characters and cannot contain emojis.
 - The name cannot start with a forward slash (/) or backslash (\). The name cannot contain consecutive forward slashes (/). You can use forward slashes (/) in a directory name to quickly create a subdirectory. For example, when you create a directory named example/test/, the directory named example/ is created in the root directory of the bucket and the subdirectory named test/ is created in the example/ directory.
 - The name cannot be two consecutive periods (...).
 - The directory name must be 1 to 254 characters in length.

5 Click OK

After you create a directory, you can click or Permanently Delete in the Actions column corresponding to the directory to delete the directory and the data in the directory.

2.5.5. Download objects

After you upload objects to a bucket, you can download the objects to the default download path of your browser or a specified local path.

Procedure

1. Log on to the OSS console.

- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the **Files** tab, and then download single or multiple objects.
- Download a single object

Method 1: Choose More > Download in the Actions column that corresponds to the object that you want to download.

Method 2: Click the name of the object that you want to download, or click **View Details** in the Actions column that corresponds to the object you want to download. In the **View Details** panel, click **Download**.

Download multiple objects

Select the objects that you want to download. Then, choose **Batch Operation > Download**. You can download up to 100 objects at a time in the OSS console.

2.5.6. Delete objects

You can delete uploaded objects in the OSS console when they are no longer needed.

Background information

You can delete a single object or batch delete multiple objects. You can batch delete up to 100 objects. To delete specific objects or batch delete more than 100 objects, we recommend that you use API operations or OSS SDKs.

() **Important** Deleted objects cannot be recovered. Exercise caution when you delete objects.

Procedure

1. Log on to the OSS console.

- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the **Files** tab, and then delete a single or multiple objects.
 - Delete a single object
 - Choose More > Permanently Delete in the Actions column corresponding to the target object.
 - Delete multiple objects at a time
 - Select the objects that you want to delete, and then choose **Batch Operation > Permanently Delete**.

4. In the message that appears, click **OK**.

2.5.7. Manage parts

When you use multipart upload to upload an object, the object is split into several parts. After all of the parts are uploaded to the OSS server, you can call CompleteMultipartUpload to combine the parts into a complete object.

Background information

You can also configure lifecycle rules to clear parts that are not needed on a regular basis. For more information, see Configure lifecycle rules.

Procedure

- 1. View publishing history.
- 2. In the left-side navigation pane, click Buckets. On the Buckets page, click the name to which you want to upload objects.
- 3. Click the Files tab. On the page that appears, click Parts.
- 4. In the **Parts** panel, delete the parts.
 - To delete all parts in the bucket, select all parts and then click Delete All.
 - To delete specific parts in the bucket, select these parts and then click **Delete**.
- 5. In the message that appears, click **OK**.

2.5.8. Configure object tagging

You can configure object tagging to classify objects. Object tagging uses key-value pairs to identify objects. You can perform operations on multiple objects that have the same tag. For example, you can configure lifecycle rules for objects that have the same tag.

Background information

Object tagging uses key-value pairs to identify objects. You can manage multiple objects that have the same tag. For example, you can configure lifecycle rules for objects that have the same tag or authorize Resource Access Management (RAM) users to access objects that have the same tag. When you configure object tagging, take note of the following items:

- A maximum of 10 tags can be configured for each object. The tags of the same object must have unique tag keys.
- A tag key can be up to 128 bytes in length. A tag value can be up to 256 bytes in length.
- Tag keys and tag values are case-sensitive.
- The key and value of a tag can contain letters, digits, spaces, and the following characters:

+ - = . _ : /

- Only the bucket owner and authorized users have read and write permissions on object tags. These permissions are independent of object access control lists (ACLs).
- In cross-region replication (CRR), object tags are also replicated to the destination bucket.

Procedure

1. Log on to the OSS console.

2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket that you want to manage.

3. In the left-side navigation pane, click Files.

- 4. Choose **More** > **Tagging** in the Actions column corresponding to the object to which you want to add tags.
- In the Tagging panel, configure the Key and Value of the tag. You can click Add to add up to more 10 tags to the object.

6. Click Confirm.

2.5.9. Configure bucket policies to authorize other users to access OSS

resources

You can configure bucket policies to grant permissions to other users to access specified Object Storage Service (OSS) resources.

Background information

You can configure multiple bucket policies for a bucket. The total size of the policies cannot exceed 16 KB.

Method 1: Configure bucket policies by using the GUI

1. Log on to the OSS console.

2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure bucket policies.

3. On the bucket details page, click the Files tab, and then click Authorize.

4. On the GUI tab, click Authorize.

5. In the Authorize panel, configure the parameters and click OK. The following table describes the parameters.

Parameter	Description
Applied To	 Select the resources on which you want to grant other users the access permissions. Whole Bucket: The bucket policy applies to all resources in the bucket. Specified Resource: The bucket policy applies only to specified resources in the bucket. You can configure multiple bucket policies for specific resources in a bucket. Configure a bucket policy for a directory To configure a bucket policy to grant users the permissions to access all subdirectories and objects in a directory, add an asterisk (*) after the name of the directory. For example, to grant users the permissions to access all subdirectories and objects in a directory named abc, enter abc/*. Configure a bucket policy for a specific object To configure a bucket policy to grant users the permissions to access a specific object, enter the full path of the object that excludes the bucket name. For example, to grant users the permissions to access an object named myphoto.png in the abc directory, enter abc/*.
Accounts	 Select the type of accounts to which you want to grant the permissions. Anonymous Accounts (*): Select this option if you want to grant all users the permissions to access the specified resources. Other Accounts: Select this option if you want to grant other Alibaba Cloud accounts, RAM users, or temporary users generated by Security Token Service (STS) the permissions to access the specified resources. To grant other Alibaba Cloud accounts or RAM users the permissions to access the specified resources, enter the UIDs of the Alibaba Cloud accounts or RAM users. To grant temporary users generated by STS the permissions to access the specified resources, enter the user and role information in the following format: arr:sts::(RoleOwnerUid):assumd=role/(RoleName)/(RoleSessionName). For example, the role used to generate a temporary user is testrole, the UID of the Alibaba Cloud account hat assumes the role is 12345, and the RoleSessionName that is specified when the temporary users the permissions to access the specified resources, use asterisks (*) as wildcard characters. For example, enter arr:sts::*:*/*/*. Important If an authorized user is a user that uses a temporary token generated by STS, the user cannot access the specified resources on the OSS console.
Authorized Operation	 You can use the following methods to specify authorized operations: Basic Settings and Advanced Settings. Basic Settings If you select this option, you can configure the following permissions based on your requirements. You can move the pointer over the icon on the right side of each permission to view the actions that correspond to the permission option. Read Only: Authorized users can view, list, and download the specified resources. Read/Write: Authorized users can read data from and write data to the specified resources. Any Operation: allows authorized users to perform all operations on the specified resources. None: Authorized users cannot perform operations on the specified resources. None: Authorized users cannot perform operations on the specified resources. Mowever, the policy in which Authorized Operation is set to None takes precedence. For example, if you configure a policy to grant the Read Only permission to a user, and then configure another policy to grant theRead/Write permission to the user is Read/Write. If you configure a third policy to grant theRead/Write permission to the user, the permission of the user is Read/Write. If you configure a third policy to grant theRead/Write permission to the user, the permission of the user is Read/Write. If you configure a third policy to grant theRead/Write permission to the user, the permission of the user is Read/Write. If you configure a third policy to grant theRead/Write permission to the user, the permission of the user is Read/Write. If you configure the following parameters: Effect: Select Allow or Deny. Action: Specify the action that you want to allow or deny. You can specify an action that is supported by OSS.

	Optional. You can configure this parameter in both Basic Settings and Advanced Settings to specify the conditions that users must meet before the users can access OSS resources.
	Access Method: Select HTTPS or HTTP.
Conditions	• IP =: Specify the IP addresses or CIDR blocks that can be used to access OSS resources. Separate multiple IP addresses with commas (,).
	• IP ≠: Specify the IP addresses or CIDR blocks that cannot be used to access OSS resources. Separate multiple IP addresses with commas (,).

6. Click Confirm.

Method 2: Configure bucket policies by specifying policy syntax

- 1. View publishing history.
- 2. In the left-side navigation pane, click **Buckets**. On the Buckets page, click the name of the bucket for which you want to configure bucket policies.
- 3. In the left-side navigation pane, click Files. On the page that appears, click Authorize.
- 4. On the Syntax tab, click Edit.
- You can specify policy syntax based on your business requirements for fine-grained access control. The following sample code provides examples on how the resource owner whose UID is 174649585760xxxx configures the bucket policies in various scenarios:
- $\circ~$ Example 1: Allow anonymous users to list all objects in a bucket named examplebucket.



"Action": "oss:*",
 "Principal": [
 "*"
],
 "Resource": [
 "acs:oss:*:174649585760xxxx:examplebucket"
],
 "Condition":{
 "NotIpAddress": {
 "acs:SourceIp": ["192.168.0.0/16"]
 }
 }
]
}

• Example 3: Allow a RAM user whose UID is 20214760404935xxxx only to read the hangzhou/2020 and hangzhou/2015 directories in a bucket named examplebucket.

1		
	"Statem	ment": [
	{	
		"Action": [
		"oss:GetObject",
		"oss:GetObjectAcl"
		"oss:GetObjectVersion".
		"oss:GetObjectVersionAcl"
		1,
		"Effect": "Allow",
		"Principal": [
		"20214760404935xxxx"
		1,
		"Resource": [
		"acs:oss:*:174649585760xxxx:examplebucket/hanghzou/2020/*"
		"acs:oss:*:174649585760xxxx:examplebucket/hangzhou/2015/*"
	},	
	{	
		"Action": [
		"oss:ListObjects",
		"oss:ListObjectVersions"
],
		"Condition": {
		"StringLike": {
		"oss:Prefix": [
		"hanghzou/2020/*",
		"hangzhou/2015/*"
		1
		}
		} <i>,</i>
		"Effect": "Allow",
		"Principal": [
		"20214760404935xxxx"
		1,
		"Resource": [
		"acs:oss:*:174649585760xxxx:examplebucket"
		1
	}	
],	
	"Versic	on": "1"
}		

5. Click **Save**, and then click **OK** in the message that appears.

Access authorized OSS resources

After you configure a bucket policy for a bucket, you can use the following methods to access the resources specified in the bucket policy:

Object URL (only for authorized anonymous users)

Anonymous users can enter the URL of an object specified in the policy in a browser to access the object. The URL of the object consists of the default domain name of the bucket and the path of the object. Example: http://mybucket.oss-cn-beijing.aliyuncs.com/file/myphoto.png .

OSS console

Log on to the OSS console. In the left-side navigation pane, click the + icon next to **My OSS Paths**. In the Add Path panel, add the bucket name and the object path specified in the bucket policy. For more information, see Add OSS paths.

2.6. Add OSS paths

You can add the paths of OSS resources in the console for quicker access.

Prerequisites

A bucket is created. For more information, see Create buckets.

- 1. Log on to the OSS console.
- 2. Click the + icon on the right side of My OSS Paths.
- 3. In the Add Authorized OSS Path panel, add a path.
 - The following table describes the parameters in the syntax.
 - $\circ~$ Region: Select the region of the bucket in the path that you want to add.
 - File Path: Add the path of the resource that you want to access. The path is in the bucket/object-prefix format. For example, if the OSS resource that you want to access is the root folder of a bucket named example, set File Path to example. If the OSS resource that you want to access is the test folder in the root folder of the bucket named example, set File Path to example/test/.
- 4. Click Confirm.

2.7. Create single tunnels

You can create single tunnels between Object Storage Service (OSS) and a virtual private cloud (VPC) to access OSS resources from the VPC.

Prerequisites

A VPC and a vSwitch are created. For more information, see the "Create a VPC" and "Create a vSwitch" topics in Apsara Stack VPC User Guide.

② Note Even if OSS is deployed in zone-disaster recovery mode, single tunnels do not provide disaster recovery capabilities and do not support switchovers.

Procedure

- 1. Log on to the OSS console.
- 2. In the left-side navigation panel, click Single Tunnel.
- 3. On the Single Tunnel page, click Create.
- 4. On the Create Single Tunnel page, configure the parameters described in the following table.

Parameter	Required	Description
Organization	Yes	Select the organization of the VPC from which you want to access OSS resources.
Resource set	Yes	After you select an organization, the resource set is automatically selected based on the organization.
Region	Yes	After you select an organization, a region is automatically selected based on the organization.
Cluster	Yes	Select the cluster to which the single tunnel applies. The single tunnel can access only the buckets in the specified cluster.
Shared scope	Yes	Select the scope of resources that you want to share. Valid values: This resource set (default), Organization and lower-level organizations , and Organization .
Description	No	Enter the description of the single tunnel you want to create. The description cannot exceed 20 characters in length.
VPC	Yes	Select the VPC in which you want to create the single tunnel. You can also click Create VPC to create a VPC.
VSwitch	Yes	Select an available vSwitch. You can also click Create VSwitch to create a vSwitch.

5. Click Submit.

Click OK. You can view the information about the single tunnel that you created on the Single Tunnel page.

If you no longer use a single tunnel, click **Delete** in the **Actions** column of the single tunnel to delete the single tunnel.

2.8. CSG 2.8.1. What is CSG?

Cloud Storage Gateway (CSG) is a gateway service that can be deployed on Apsara Stack. CSG uses OSS buckets as backend storage devices. CSG provides on-premises and cloud applications with standard file services over the Network File System (NFS) and Server Message Block (SMB) protocols.

CSG supports only file gateways.

File gateways use OSS buckets as backend storage devices, and map the object directory structure of OSS buckets to NAS file systems. You can read and write all objects in a specified OSS bucket over standard NFS and SMB protocols. CSG also uses on-premises storage to cache hot data, and provides high-performance data access in addition to the large storage capacity of OSS buckets. File gateways are compatible with the Portable Operating System Interface (POSIX) and third-party backup software.

Service architecture

The following figure shows the architecture of CSG.



Scenarios

File gateways

- Build a file storage service for a large file system when local storage is limited.
- Store data as objects in the cloud, and allow applications to access the data in a file system without the need to modify code.

2.8.2. Usage notes

Before you run CSG instances, we recommend that you read the following usage notes.

File gateways

- You can create file gateways only by using the recommended CIDR block or its subnets of a VPC.
- You may have a large number of files exist in the same directory of an OSS bucket. For example, hundreds of thousands or even millions of
 subdirectories or files exist in the root directory or under a specific directory of an OSS bucket. In this case, file gateways are not suitable due to
 extremely slow access to subdirectories or files. In most cases, if you want to use file gateways to manage files, we recommend that you place at
 most 100,000 files in a single directory.
- Files gateways reserve 20% of the cache disk space to cache metadata and use the rest of the space to cache data. The stored metadata refers to the metadata of different files, including the size and last modification time of each file. In this case, if a file is discarded, a file gateway creates a metadata file so that you can view the information about the discarded file when you access folders from the client. If you want to read the discarded file from the OSS bucket to the cache disk. The space required to cache the metadata on the cache disk determines the maximum number of files that are supported by the current share. In most cases, a share with a cache disk of 100 GB in size supports at most 10 million files. You can check the value of the Available Metadata Space parameter of shares when you use the file gateway. Generally, you can consider whether to scale out the cache disk when the remaining space is about 3 GB.
- Files written to file gateways are asynchronously synchronized to OSS buckets. Before the files are synchronized to OSS buckets by using file gateways, we recommend that you do not modify or delete the files by using other gateways or tools, such as ossutil or ossfs. Otherwise, data inconsistency may occur.
- We recommend that you do not frequently interrupt the upload of large files to NFS or SMB shares. The system uploads files by using multipart uploads. If you interrupt the upload of large files, file fragments are generated in the associated OSS bucket. These file fragments occupy the capacity of the OSS bucket. Therefore, the storage usage of the OSS bucket is slightly higher than the total file size.
- The cache capacity for file sharing is calculated based on the following formula: Recommended local cache capacity = [Application bandwidth (Mbit/s) Backend bandwidth of a gateway (Mbit/s)] × Write duration (seconds) × 1.2.

To obtain better performance when you access data from local clients, you can estimate the total amount of hot data. Compare the total amount of hot data with the recommended on-premises cache capacity and select the higher value as the capacity of the on-premises cache disk.

- If you want to write large files by using a file gateway, the size of each file must be smaller than 30% of the cache disk capacity. You cannot write multiple large files at the same time. If you write multiple large files at the same time, the cache disk space is exhausted.
- File gateways support sparse files. If a sparse file fails to be uploaded to a file gateway, run the following command to convert the format of the sparse file:

dd if=<sparse file name> of=<sparse file name> conv=notrunc bs=1M

The size of the sparse file cannot exceed the available capacity of the cache disk.

• The names of file gateways and directories must be encoded in UTF-8. File gateways only support file and directory names that are encoded in UTF-8. Other formats are not supported. For example, if you mount an NFS file share of a file gateway on a Windows CSG agent, the files and directories that have Chinese names cannot be created. In this case, a 0x8007045D error is returned.

 If the size of a file in a file gateway exceeds 256 MB, we recommend that you disable the versioning feature for the associated OSS bucket. Otherwise, a timeout error may occur when the gateway uploads metadata to the associated bucket. This affects the performance of the gateway.

2.8.3. Limits

This topic describes the limits of CSG.

- CSG is an independently deployed cloud service. A CSG instance cannot be mapped to an OSS bucket that is associated with a virtual private cloud (VPC).
- A CSG instance cannot be mapped to an OSS bucket for which the write once read many (WORM) feature is enabled.
- CSG cannot detect the capacity threshold that is configured for an OSS bucket. When the data in an OSS bucket to which the CSG instance is mapped exceeds the capacity threshold, subsequent data written into the CSG instance cannot be synchronized to the bucket.

2.8.4. Quick start

After you use file gateways to map the objects and folders in OSS buckets to the files and directories in Apsara File Storage NAS file systems, you can read and write all objects in a specified OSS bucket by using the NFS and SMB protocols. CSG uses on-premises storage to cache hot data, and provides high-performance data access in addition to the large storage capacity of OSS buckets.

Prerequisites

- A VPC and a vSwitch are created. For more information, see the Create and manage a VPC and Create and manage a vSwitch topics in VPC User Guide.
- An OSS bucket is created. For more information, see Create buckets.
- An Elastic Compute Service (ECS) instance is created and used as a client, and the ECS instance is associated with the created VPC. For more information, see the Create an instance topic in **ECS User Guide**.

Background information

CSG is a gateway service that can be deployed at your self-managed data center or on Apsara Stack. CSG uses OSS buckets as backend storage devices. CSG provides on-premises and cloud applications with standard file services over the NFS and SMB protocols by using cost-efficient virtual machines.

Step 1: Create a gateway

- 1. Log on to the OSS console.
- 2. Click Cloud Storage Gateways in the left-side navigation pane.
- 3. If you use CSG for the first time, you must authorize CSG to access other Apsara Stack services based on your business requirements. When you perform the authorization operation, the service-linked role AliyunServiceRoleForHCSSGW is automatically created. The system attaches the AliyunServiceRolePolicyForHCSSGW policy to this service-linked role so that CSG can assume the service-linked role and access other Alibaba Cloud services, such as OSS, VPC, and ECS. CSG provides a mount protocol based on the granted permissions to manage requests to OSS buckets, such as the requests to upload, download, or access data in OSS buckets.
- i. In the Storage Gateway Service Authorization message, read the authorization information and click Authorization.
- ii. Follow the instructions as prompted to complete the authorization.
- 4. On the Gateways page, click Create.
- 5. In the Gateway Information step, set the parameters and click Next. The following table describes the parameters.

Parameter	Description
Name	The name of the gateway.
Description	The description of the gateway.

6. In the Gateway Configurations step, set the parameters and click Next. The following table describes the parameters.

Parameter	Description	
VPC	The VPC in which you want to deploy the gateway. ① Important You must select the VPC in which the ECS instance resides.	
VSwitch	The vSwitch that is connected to the gateway. ① Important You must select the vSwitch to which the ECS instance is connected.	
Gateway ECS Specifications	After you select a vSwitch, you can view the specifications of the ECS instance, the number of vCPU cores, memory size, and internal bandwidth.	

7. In the Confirmation step, check the information of the gateway and click Completed

After the gateway is created, the system automatically deploys and starts the gateway. It takes about 5 to 10 minutes to complete the process. The gateway enters the **Running** state if it is activated and deployed.

Step 2: (Optional) Create an SMB user

If you want to access SMB share directories as an SMB user, you must create an SMB user in advance.

() Note By default, if you have not created an SMB user, you can access SMB share directories as a public user. However, if you have created an SMB user, you must grant read and write or read-only permissions to the user before you access SMB share directories as an SMB user. For more information, see Configure an SMB share.

- 1. On the Gateways page, find the gateway for which you want to create an SMB user and click the ID of the gateway.
- 2. On the page that appears, click SMB Users in the left-side navigation pane.
- 3. On the SMB Users page, click Create in the upper-right corner.
- 4. In the Add SMB User dialog box, enter the username and password of the SMB user.

5. Click OK

Step 3: Create a share

- 1. On the Gateways page, find the gateway for which you want to create a share and click the ID of the gateway.
- 2. The **Shares** page appears. On the Shares page, click **Create** in the upper-right corner.
- 3. In the **Bucket Settings** step, set the parameters and click **Next**. The following table describes the parameters.

Parameter	Description	
Custom Bucket Domain	 Specifies whether to use a custom domain name for the OSS bucket. Valid values: Yes: uses a custom domain name for the OSS bucket. If you select Yes, you can access a bucket that resides in a region different from that in which the specified gateway resides. No: uses the default domain name of the OSS bucket in the current region. If you select No, you can access only a bucket that resides in the same region as the specified gateway. 	
Bucket Domain	The domain name of the bucket.	
Bucket Name	The name of the bucket.	
Subdirectory	If you select Subdirectory , enter the names of the subdirectories of the OSS bucket in the field.	
Use SSL to Connect Bucket	 Specifies whether to connect to the OSS bucket by using SSL. Valid values: Yes: uses SSL to connect to the OSS bucket. This is the default value. No: does not use SSL to connect to the OSS bucket. 	
Bucket AccessKey ID	The AccessKey ID and AccessKey secret that are used to access the OSS bucket when you specify a custom domain name for the OSS bucket. This parameter is required only if you use a custom domain name of the	
Bucket AccessKey Secret	OSS bucket. () Important The account to which the AccessKey ID and AccessKey secret belong must have the required permissions to access the OSS bucket. For more information about the permissions on OSS buckets, see Permissions required by a gateway to operate OSS buckets	

4. In the Basic Information step, set the parameters and click Next. The following table describes the parameters.

Parameter	Description
Share Name	The name of the NFS or SMB share that you want to create. If you set the Protocol parameter to NFS, the share name also specifies the virtual path of NFS version 4 (NFSv4). The name can be up to 32 characters in length and can contain letters and digits. The name must start with a letter.
Protocol	The type of protocol used by the share. Valid values: • NFS : The NFS protocol is suitable if you need to access OSS buckets from a Linux operating system. • SMB : The SMB protocol is suitable if you need to access OSS buckets from a Windows operating system.
Cache	Select Create Cache to create a cache disk.
Cache Type	The type of the cache disk. Valid values include Ultra Disk and SSD . The valid values vary based on the backend storage that you use.
Cache Size	The capacity of the cache disk. Valid values: 100 to 4096.
Cache	Select an existing cache disk. If no cache disk is available, select Create Cache and then set the Cache Type and Cache Size parameters. Note A proportion of 20% of the cache disk space is used to cache metadata.
User Mapping	 Maps an NFS client user to an NFS server user. You can set this parameter only if you set thProtocol parameter to NFS. Valid values: none: NFS client users are not mapped to the nobody user on the NFS server. root_squash: restricts the use of root user permissions. NFS clients that use the root identity are mapped to the nobody user on the NFS server. all_squash: restricts all user permissions. The NFS client is mapped to the nobody user on the NFS server regardless of the identity that the client uses. all_anonymous: restricts all user permissions. The NFS client is mapped to the anonymous user on the NFS server regardless of the identity that the client uses.
Browsable	 Specifies whether the SMB share can be discovered by network neighbors. You can set this parameter only if you set the Protocol parameter to SMB. Valid values: Yes: The SMB share can be discovered by network neighbors. No: The SMB share cannot be discovered by network neighbors.
Advanced Settings	To configure advanced settings such as reverse sync, select Advanced Settings.

 In the Advanced Settings step, set the required parameters, and click Next. The following table describes the parameters. You can configure settings in the Advanced Settings step only if you select Advanced Settings in the Basic Information step.

If an object exists in the current OSS bucket, the reverse sync feature is automatically enabled. If you want to disable this feature, you can set Reverse Sync to No in the Advanced Settings step. After you disable the reverse sync feature, objects that exist in the OSS bucket before you mount the share cannot be discovered by the gateway.

Parameter

Description

	The data storage mode. Valid values:
	 Replication Mode: In this mode, two backups are created for all data. One backup is stored in the on- premises cache disk, and the other backup is stored in the associated OSS bucket.
	In this mode, the total size of files that can be written to the gateway is equal to the specified size of the cache disk.
mode	 Cache Mode: In this mode, the backup that is stored in the on-premises cache disk contains only metadata and the user data that is frequently accessed. Full data is stored in the OSS.
	In this mode, the total size of files that can be written to the gateway is not affected by the size of the cache disk. We recommend that you set Mode to Cache Mode if you want to migrate data to the cloud or back up data.
	Specifies whether to enable the reverse sync feature to synchronize metadata from the OSS bucket to the on- premises cache disk. This setting is applicable to disaster recovery, data restoration, and data sharing scenarios. Valid values:
	• Yes: enables the reverse sync feature.
Reverse Sync	If you enable the reverse sync feature, the gateway scans all objects in the OSS bucket. The scanning process is implemented by calling the API operations provided by OSS. If a large number of objects exist in the OSS bucket, we recommend that you specify a time interval of at least 600 seconds to prevent low efficiency.
	No: disables the reverse sync feature.
Reverse Sync Interval	The interval between two consecutive reverse synchronization tasks. Valid values: 15 to 36000. Unit: seconds. You can set this parameter only if you set the Reverse Sync parameter to Yes .
Ignore Deletions	If you select Yes, the data that is deleted from the on-premises cache disk is not deleted from the OSS bucket. Full data is stored in the OSS.
Sync Latency	The latency for synchronizing files from the on-premises cache disk to the OSS bucket. Default value: 5. Maximum value: 120. Unit: seconds.
	After you set this parameter, the gateway does not upload the files that you modified and closed until the specified amount of time is reached. This prevents file fragments generated in the OSS bucket due to frequent on-premises modifications.

6. In the Confirmation step, check the information of the share and click Completed.

What to do next

After you create a shared directory, you can access the shared directory. For more information, see Access an NFS share and Access an SMB share.

2.8.5. File gateways

2.8.5.1. Manage file gateways

This topic describes how to manage file gateways. You can follow the instructions provided in this topic to create, modify, and delete a file gateway.

Prerequisites

- A VPC and a vSwitch are created. For more information, see the Create and manage a VPC and Create and manage a vSwitch topics in VPC User Guide.
- An Elastic Compute Service (ECS) instance is created and used as a client, and the ECS instance is associated with the created VPC. For more information, see the Create an instance topic in **ECS User Guide**.

Create a gateway

1. Log on to the OSS console.

- 2. Click **Cloud Storage Gateways** in the left-side navigation pane.
- 3. If you use CSG for the first time, you must authorize CSG to access other Apsara Stack services based on your business requirements. When you perform the authorization operation, the service-linked role AliyunServiceRoleForHCSSGW is automatically created. The system attaches the AliyunServiceRolePolicyForHCSSGW policy to this service-linked role so that CSG can assume the service-linked role and access other Alibaba Cloud services, such as OSS, VPC, and ECS. CSG provides a mount protocol based on the granted permissions to manage requests to OSS buckets, such as the requests to upload, download, or access data in OSS buckets.
- i. In the Storage Gateway Service Authorization message, read the authorization information and click Authorization
- ii. Follow the instructions as prompted to complete the authorization.

4. On the Gateways page, click Create.

5. In the Gateway Information step, set the parameters and click Next. The following table describes the parameters.

Parameter	Description
Name	The name of the gateway.
Description	The description of the gateway.

6. In the Gateway Configurations step, set the parameters and click Next. The following table describes the parameters.

Parameter	Description
VPC	The VPC in which you want to deploy the gateway. ① Important You must select the VPC in which the ECS instance resides.
VSwitch	The vSwitch that is connected to the gateway. Important You must select the vSwitch to which the ECS instance is connected.
Gateway ECS Specifications	After you select a vSwitch, you can view the specifications of the ECS instance, the number of vCPU cores, memory size, and internal bandwidth.

7. In the Confirmation step, check the information of the gateway and click Completed. After the gateway is created, the system automatically deploys and starts the gateway. It takes about 5 to 10 minutes to complete the process. The gateway enters the Running state if it is activated and deployed.

More operations

After a gateway is deployed, you can view the information about the gateway on the Gateways page, such as the service IP address, type, and status of the gateway. You can also perform the following operations on a gateway as needed.

Operation	Description
Modify a gateway	Find the gateway that you want to modify and click Edit in the Actions column. Then, you can modify the name and description of the gateway.
Update the version of a gateway	By default, each gateway that you create is of the latest version. You can update a gateway only if a later gateway version is available. Find the gateway whose version you want to update and choose More > Upgrade in the Actions column.
Restart a gateway	Find the gateway that you want to restart and choose More > Restart Gateway in the Actions column.
Delete a gateway	Find the gateway that you want to delete and choose More > Delete in the Actions column.

After a gateway is deployed, click the name of the gateway on the Gateways page. Then, you can perform the following operations on different pages based on your business requirements.

- On the Share page, you can create a share and restart NFS shares or SMB shares. For more information, see Manage shares.
- On the Details page, you can view the basic information about the gateway, including the name, ID, version, type, status, creation time, and activation time.
- On the Cache page, you can view the information about the cache disks and scale out a cache disk. For more information, see Manage cache disks.
- On the SMB Users page, you can create and delete SMB users. For more information, see Create an SMB users.

What to do next

Manage shares

2.8.5.2. Manage shares

This topic describes how to manage shares in the CSG console. For example, you can create, delete, and configure NFS and SMB shares.

Prerequisites

- A gateway is created. For more information, see Create a gateway.
- An OSS bucket is created. For more information, see Create buckets.

Create a share

1. Log on to the OSS console.

- 2. Click Cloud Storage Gateways in the left-side navigation pane.
- 3. On the Gateways page, find the gateway for which you want to create an SMB user and click the ID of the gateway.
- 4. The Shares page appears. On the Shares page, click Create in the upper-right corner.
- 5. In the Bucket Settings step, set the parameters and click Next. The following table describes the parameters.

Parameter	Description
Custom Bucket Domain	 Specifies whether to use a custom domain name for the OSS bucket. Valid values: Yes: uses a custom domain name for the OSS bucket. If you select Yes, you can access a bucket that resides in a region different from that in which the specified gateway resides. No: uses the default domain name of the OSS bucket in the current region. If you select No, you can access only a bucket that resides in the same region as the specified gateway.
Bucket Domain	The domain name of the bucket.
Bucket Name	The name of the bucket.
Subdirectory	If you select Subdirectory , enter the names of the subdirectories of the OSS bucket in the field.
Use SSL to Connect Bucket	 Specifies whether to connect to the OSS bucket by using SSL. Valid values: Yes: uses SSL to connect to the OSS bucket. This is the default value. No: does not use SSL to connect to the OSS bucket.
Bucket AccessKey ID	The AccessKey ID and AccessKey secret that are used to access the OSS bucket when you specify a custom domain name for the OSS bucket. This parameter is required only if you use a custom domain name of the
Bucket AccessKey Secret	OSS bucket. () Important The account to which the AccessKey ID and AccessKey secret belong must have the required permissions to access the OSS bucket. For more information about the permissions on OSS buckets, see Permissions required by a gateway to operate OSS buckets

6. In the Basic Information step, set the parameters and click Next. The following table describes the parameters.

Parameter	Description				
Share Name	The name of the NFS or SMB share that you want to create. If you set the Protocol parameter to NFS, the share name also specifies the virtual path of NFS version 4 (NFSv4). The name can be up to 32 characters in length and can contain letters and digits. The name must start with a letter.				
Protocol	The type of protocol used by the share. Valid values: NFS: The NFS protocol is suitable if you need to access OSS buckets from a Linux operating system. SMB: The SMB protocol is suitable if you need to access OSS buckets from a Windows operating system. 				
-------------------	--	--	--	--	--
Cache	Select Create Cache to create a cache disk.				
Cache Type	The type of the cache disk. Valid values include Ultra Disk and SSD . The valid values vary based on the backend storage that you use.				
Cache Size	The capacity of the cache disk. Valid values: 100 to 4096.				
Cache	Select an existing cache disk. If no cache disk is available, select Create Cache and then set the Cache Type and Cache Size parameters. Note A proportion of 20% of the cache disk space is used to cache metadata.				
User Mapping	 Maps an NFS client user to an NFS server user. You can set this parameter only if you set thProtocol parameter to NFS. Valid values: none: NFS client users are not mapped to the nobody user on the NFS server. root_squash: restricts the use of root user permissions. NFS clients that use the root identity are mapped to the nobody user on the NFS server. all_squash: restricts all user permissions. The NFS client is mapped to the nobody user on the NFS server regardless of the identity that the client uses. all_anonymous: restricts all user permissions. The NFS client is mapped to the anonymous user on the NFS server regardless of the identity that the client uses. 				
Browsable	 Specifies whether the SMB share can be discovered by network neighbors. You can set this parameter only if you set the Protocol parameter to SMB. Valid values: Yes: The SMB share can be discovered by network neighbors. No: The SMB share cannot be discovered by network neighbors. 				
Advanced Settings	To configure advanced settings such as reverse sync, select Advanced Settings.				

 In the Advanced Settings step, set the required parameters, and click Next. The following table describes the parameters. You can configure settings in the Advanced Settings step only if you select Advanced Settings in the Basic Information step.

If an object exists in the current OSS bucket, the reverse sync feature is automatically enabled. If you want to disable this feature, you can set Reverse Sync to No in the Advanced Settings step. After you disable the reverse sync feature, objects that exist in the OSS bucket before you mount the share cannot be discovered by the gateway.

Parameter	Description
Mode	 The data storage mode. Valid values: Replication Mode: In this mode, two backups are created for all data. One backup is stored in the on-premises cache disk, and the other backup is stored in the associated OSS bucket. In this mode, the total size of files that can be written to the gateway is equal to the specified size of the cache disk. Cache Mode: In this mode, the backup that is stored in the on-premises cache disk contains only metadata and the user data that is frequently accessed. Full data is stored in the OSS. In this mode, the total size of files that can be written to the gateway is not affected by the size of the cache disk. We recommend that you set Mode to Cache Mode if you want to migrate data to the cloud or back up data.
Reverse Sync	 Specifies whether to enable the reverse sync feature to synchronize metadata from the OSS bucket to the on-premises cache disk. This setting is applicable to disaster recovery, data restoration, and data sharing scenarios. Valid values: Yes: enables the reverse sync feature. If you enable the reverse sync feature, the gateway scans all objects in the OSS bucket. The scanning process is implemented by calling the API operations provided by OSS. If a large number of objects exist in the OSS bucket, we recommend that you specify a time interval of at least 600 seconds to prevent low efficiency. No: disables the reverse sync feature.
Reverse Sync Interval	The interval between two consecutive reverse synchronization tasks. Valid values: 15 to 36000. Unit: seconds. You can set this parameter only if you set the Reverse Sync parameter to Yes .
Ignore Deletions	If you select Yes, the data that is deleted from the on-premises cache disk is not deleted from the OSS bucket. Full data is stored in the OSS.
Sync Latency	The latency for synchronizing files from the on-premises cache disk to the OSS bucket. Default value: 5. Maximum value: 120. Unit: seconds. After you set this parameter, the gateway does not upload the files that you modified and closed until the specified amount of time is reached. This prevents file fragments generated in the OSS bucket due to frequent on-premises modifications.

8. In the **Confirmation** step, check the information of the share and click **Completed**.

Configure an NFS share

If you select NFS as Protocol when you create a share, you can configure an NFS share. For example, you can set the Read/Write Users and Read-only Users parameters.

1. On the **Shares** page, find the share that you want to configure and click **Settings** in the Actions column.

2. In the NFS Share Settings dialog box, set the parameters. The following table describes the parameters.

Parameter

Description

		Maps an NFS client user to an NFS server user. Valid values:
		 none: NFS client users are not mapped to the nobody user on the NFS server.
		 root_squash: restricts the use of root user permissions. NFS clients that use the root identity are mapped to the nobody user on the NFS server.
	User Mapping	 all_squash: restricts all user permissions. The NFS client is mapped to the nobody user on the NFS server regardless of the identity that the client uses.
		 all_anonymous: restricts all user permissions. The NFS client is mapped to the anonymous user on the NFS server regardless of the identity that the client uses.
	Read/Write Clients	The IP addresses or CIDR blocks of the NFS share that allow read and write access. Examples: 192.168.10.10 and 192.168.0.0/24. You can enter multiple IP addresses or CIDR blocks.
	Read-only Clients	The IP addresses or CIDR blocks of the clients that can only read data from the NFS share. Examples: 192.168.10.10 and 192.168.0.0/24. You can enter multiple IP addresses or CIDR blocks.
	Write Speed Limit	The maximum write speed. Valid values: 0 to 1280. Unit: Mbit/s. Default value: 0. The default value 0 indicates that the write speed is unlimited.
		The maximum upload speed. Valid values: 0 to 1280. Unit: Mbit/s. Default value: 0. The default value 0 indicates that the upload speed is unlimited.
Upload Speed Limit	Upload Speed Limit	(?) Note When you set the maximum write speed and upload speed, make sure that the maximum upload speed is not lower than the maximum write speed.

3. Click **OK**.

Configure an SMB share

If you select SMB as Protocol when you create a share, you can configure an SMB share. For example, you can set the Read/Write Users and Read-only Users parameters.

1.	On the Shares page,	find the share that	you want to	configure and c	click Settings in	the Actions column.
----	----------------------------	---------------------	-------------	-----------------	--------------------------	---------------------

2.	In the SMB Share Settings dialog box	, set the parameters	. The following tak	ole describes the parameters	;.

Parameter	Description
Browsable	 Specifies whether the SMB share can be discovered by network neighbors. Valid values: Yes: The SMB share can be discovered by network neighbors. No: The SMB share cannot be discovered by network neighbors.
Read/Write Users	The users who are allowed to read and write data from and to the SMB share.
Read-only Users	The users who are allowed to only read data from the SMB share. ⑦ Note If you grant both the read-only and read/write permissions to a user, the read-only permission takes effect.
Write Speed Limit	The maximum write speed. Valid values: 0 to 1280. Unit: Mbit/s. Default value: 0. The default value 0 indicates that the write speed is unlimited.
Upload Speed Limit	The maximum upload speed. Valid values: 0 to 1280. Unit: Mbit/s. Default value: 0. The default value 0 indicates that the upload speed is unlimited. Note When you set the maximum write speed and upload speed, make sure that the maximum upload speed is not lower than the maximum write speed.

What to do next

On the **Shares** page, you can perform the operations described in the following table.

Operation	Description				
Modify the advanced settings of a share	Find the share for which you want to modify the advanced settings and click Advanced Settings in the Actions column. In the dialog box that appears, modify the advanced settings of the share. For example, you can modify the Reverse Sync and Ignore Deletions parameters. For more information, see Create a share .				
	Find the share that you want to delete and click Delete in the Actions column.				
Delete a share	 Note This operation does not delete the data stored in the associated OSS bucket. This operation does not release the attached cache disk. If you create another share, you must attach a cache disk and an OSS bucket to the share. 				
Restart NFS shares	Click Restart NFS Shares to restart all the NFS shares that are connected to the current gateway.				
Restart SMB shares	Click Restart SMB Shares to restart all the SMB shares that are connected to the current gateway.				
Hide tasks	Click Hide Tasks in the upper part of the page to hide the tasks at the bottom of the page.				
View the upload and download queues	 Find the share whose upload and download queues you want to view, and click the plus sign (+) next to the share name to view the upload and download queues. If the number of objects in the upload queue is not 0, one or more objects are waiting to be uploaded to the associated OSS bucket. If the number of objects in the download queue is not 0, one or more objects are waiting to be downloaded. If the numbers of files in the upload and download queues are both 0, data is synchronized between the gateway and the OSS bucket. 				

2.8.5.3. Manage cache disks

This topic describes how to manage cache disks. You can scale up or delete a cache disk.

Prerequisites

A share is created. For more information, see Create a share.

Background information

Each share of a file gateway is attached a unique cache disk. To create multiple shares, you must create the same number of cache disks for the shares. By using a cache disk, you can upload data from a share to an Object Storage Service (OSS) bucket or download data from an OSS bucket to a local device.

Scale up a cache disk

① Important If you scale up a cache disk, the share to which the cache disk is attached may be unavailable for about 5 minutes. After the cache disk is scaled up, the share is automatically recovered.

If a cache disk runs out of space, you can scale up the cache disk.

1. Log on to the OSS console

- 2. Click Cloud Storage Gateways in the left-side navigation pane.
- 3. On the Gateways page, find the gateway for which you want to create an SMB user and click the ID of the gateway.
- 4. Click the **Cache** tab. On the Cache tab, find the cache disk that you want to scale up and click the plus icon.
- In the Scale Up Cache dialog box, set the Capacity parameter to expand the capacity of the cache disk. Valid values of the Capacity parameter: 100 to 2048. Unit: GB.
- 6. Click OK.

Delete a cache disk

You can delete a cache disk only if the cache disk is attached to no share.

- 1. Log on to the OSS console.
- 2. Click Cloud Storage Gateways in the left-side navigation pane.
- 3. On the Gateways page, find the gateway for which you want to create an SMB user and click the ID of the gateway.
- 4. Click the Cache tab. On the Cache tab, find the cache disk that you want to delete and click the cice.

5. In the message that appears, click **OK**.

2.8.5.4. Create an SMB user

This topic describes how to create a Server Message Block (SMB) user. SMB users can access SMB shares.

Procedure

- 1. Log on to the OSS console.
- 2. Click Cloud Storage Gateways in the left-side navigation pane.
- 3. On the Gateways page, find the gateway for which you want to create an SMB user and click the ID of the gateway.
- 4. Click the SMB Users tab. On the SMB Users tab, click Create in the upper-right corner.
- 5. In the Add SMB User dialog box, specify the username and password of the SMB user that you want to create.
- 6. Click OK.

2.8.5.5. Access shares

2.8.5.5.1. Access an NFS share

This topic describes how to access a Network File System (NFS) share of a gateway from a client that is installed on a Linux Elastic Compute Service (ECS) instance.

Prerequisites

An NFS share is created. For more information, see Create a share.

(? Note You can mount up to eight NFS or Server Message Block (SMB) shares to a gateway.

Manually mount an NFS share

- 1. Log on to the ECS console. For more information, see the Log on to the ECS console topic in ECS User Guide.
- 2. Connect to your Linux ECS instance. For more information, see the Connect to an instance topic in ECS User Guide.
- 3. On the ECS instance, run the following command to mount an NFS share to a local directory of the client:

mount.nfs 192.168.0.0:/shares local-directory

Parameter description:

- 192.168.0.0:/shares : the mount target of the gateway. The mount target consists of the IPv4 address of the gateway and the name of the NFS share. Specify the mount target based on your business requirements. To view the mount target of the gateway, go to the Share tab of the gateway
- local-directory : a local directory of the client. Specify a directory that supports read and write operations. You cannot specify a directory that does not exist.
- 4. Run the df -h command to check the mount result.

If information similar to the following output appears, the NFS share is mounted to the local directory of the client.

<pre>[root@centos7cb ~]#</pre>	df -h				
Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/vda1	99G	1.6G	92G	2%	/
devtmpfs	24G	Θ	24G	0%	/dev
tmpfs	24G	Θ	24G	0%	/dev/shm
tmpfs	24G	424K	24G	1%	/run
tmpfs	24G	Θ	24G	0%	/sys/fs/cgroup
tmofe	1 80	0	1 80	<u>0</u> %	/run/usor/A
l :/nfs2	256T	õ	256T	0%	/mnt/nfs172cent7.4
[root@centos7cb ~]#					

③ Note After the NFS share is mounted, you can view the capacity of the file system that is managed by the NFS share.

Access an NFS share

After an NFS share is mounted, you can access the NFS share in the same way that you access a local directory. If you have write permissions on the NFS share, you can write data to the NFS share. If you have read-only permissions on the NFS share, you can only read data from the NFS share.

(2) Note A share of a gateway is synchronized with the Object Storage Service (OSS) bucket associated with the share. When you manage a share, the changes to the share are also applied to the associated OSS bucket.

Enable automatic mounting of an NFS share

When you restart the ECS instance on which an NFS file system is mounted, the mount information about the file system may be lost. To prevent this issue, you can modify the /etc/fstab or /etc/rc.local configuration file on the Linux ECS instance to enable automatic mounting at system startup for the NFS file system. We recommend that you use the /etc/fstab file.

③ Note Before you enable automatic mounting, make sure that the preceding manual mounting is successful. This prevents startup failures of the ECS instance.

1. Open a configuration file and add the mount command.

• Method 1 (recommended): Open the /etc/fstab file and add the mount command.

(?) Note If you enable automatic mounting in CentOS 6, perform the following steps first:

- a. Run the chkconfig netfs on command to enable the netfs service to start at system startup.
- b. Open the /etc/netconfig file and comment out inet6-related information.
- To use NFSv4 to mount the file system, add the following command:

192.168.0.0:/shares local-directory nfs defaults 0 0

To use NFSv3 to mount the file system, add the following command:

192.168.0.0:/shares local-directory nfs vers=3.0 defaults 0 0

• Method 2: Open the /etc/rc.local file and add the mount command.

Note Before you modify the /etc/rc.local file, make sure that you have execute permissions on the /etc/rc.local and /etc/rc.local files. For example, in CentOS 7.x, execute permissions are not granted by default. Before you modify the /etc/rc.local file, grant execute permissions on the files to the account that you use to log on to the ECS instance.

To use NFSv4 to mount the file system, add the following command:

sudo mount.nfs 192.168.0.0:/shares local-directory

• To use NFSv3 to mount the file system, add the following command:

sudo mount -t nfs -o vers=3,proto=tcp,nolock 192.168.0.0:/shares local-directory

Parameter description:

- 192.168.0.0:/shares : the mount target of the gateway. The mount target consists of the IPv4 address of the gateway and the name of the NFS share. Specify the mount target based on your business requirements. To view the mount target of the gateway, go to the **Share** tab of the gateway.
- local-directory is a local directory of the client. Specify a directory that supports read and write operations. You cannot specify a directory that does not exist.

2. Run the **reboot** command to restart the ECS instance.

2.8.5.5.2. Access an SMB share

This topic describes how to access a Server Message Block (SMB) share of a gateway from a client that is installed on a Windows Elastic Compute Service (ECS) instance.

Prerequisites

An SMB share is created. For more information, see Create a share.

User Guide-OSS

? Note

• You can mount up to eight Network File System (NFS) or SMB shares to a gateway.

If you have not created an SMB user, you can access SMB shares from a client as a public user by default. If you have created an SMB user, you must grant read and write or read-only permissions to the user before the user can access SMB shares. For more information, see Configure an SMB share.

After you change the permissions of an SMB user, run the net use /delete < share path > command to delete the share in the Windows
operating system. You do not need to restart the client.

Procedure

- 1. Log on to the ECS console. For more information, see the Log on to the ECS console topic in ECS User Guide.
- 2. Connect to your Windows ECS instance. For more information, see the Connect to an instance topic in ECS User Guide.
- 3. Open This PC and click Map network drive.
- 4. Select a drive letter from the drop-down list and enter the mount target of the gateway in the Folder field.

The mount target consists of the IP address of the gateway and the name of the SMB share. Specify the mount target based on your business requirements. To view the mount target of the gateway, go to the **Share** tab of the gateway. By default, a mount target uses IPv4.

- 5. Click Finish and enter the username and password of the SMB user.
- 6. After you mount the SMB share, check the mount result.

If a window similar to the following one appears, the SMB share is mounted.

👳 🗹 📙 🖛	Drive Tools smb (\\` 3) (Z	:)					-		х
File Home Shar	re View Manage								~ ?
← → ~ ↑ 至 > T	This PC → smb (\\172.16.0.68) (Z:)					∿ ©	Search smb (\\`) (Z:)	P
	Name	Date modified	Туре	Size					
Desktop	test.txt	10:39 AM	Text Document		0 KB				
🚽 Downloads 🚿									
🔮 Documents 🚿	*								
📰 Pictures 🛛 🖈	•								
💻 This PC									
E. Desktop									
Documents									
Downloads									
J Music									
Pictures									
Videos									
🏪 Local Disk (C:)									
🛫 smb (\\' 3	0								
i Network									

7. Access the SMB share

After an SMB share is mounted, you can access the SMB share in the same way that you access a local directory. If you have write permissions on the SMB share, you can write data to the SMB share. If you have read-only permissions on the SMB share, you can only read data from the SMB share. For more information about permissions on an SMB share, see Configure an SMB share.

③ Note A share of a gateway is synchronized with the Object Storage Service (OSS) bucket associated with the share. When you manage a share, the changes to the share are also applied to the associated OSS bucket.

2.8.5.6. Update a gateway

By default, each gateway that you create is of the latest version. You can update a gateway only if a later gateway version is available. This topic describes how to update a gateway.

Procedure

- 1. Log on to the OSS console.
- 2. Click Cloud Storage Gateways in the left-side navigation pane.
- 3. On the Gateways page, find the gateway that you want to update and click Upgrade in the Actions column.

② Note During the update, the gateway may fail to respond to requests from clients.

2.8.5.7. Appendix

2.8.5.7.1. Permissions required by a gateway to operate OSS buckets

Scenario in which a share is created by using a default OSS bucket domain name

If you create a share for a gateway by using a default Object Storage Service (OSS) bucket domain name, you do not need to grant permissions on the OSS bucket to the gateway. The permissions required by gateways to operate default OSS buckets are granted when you authorize Cloud Storage Gateway (CSG) to access your cloud resources.

Scenario in which a share is created by using a custom OSS bucket domain name

If you create a share for a gateway by using a custom OSS bucket domain name, the Alibaba Cloud account to which the input AccessKey ID and AccessKey secret belong must be granted the permissions to perform the following actions on the OSS bucket:

"Action": [
 "oss:ListBuckets",
 "oss:ListObjects",
 "oss:GetObject",
 "oss:DeleteObject",
 "oss:DeleteObject",
 "oss:HeadObject",
 "oss:CopyObject",
 "oss:InitiateMultipartUpload",
 "oss:OploadPartCopy",
 "oss:CompleteMultipartUpload",
 "oss:ListMultipartUploads",
 "oss:ListParts",

"oss:ListParts", "oss:GetBucketStat", "oss:GetBucketWebsite", "oss:GetBucketInfo", "oss:GetBucketEncryption", "oss:GetBucketVersioning", "oss:PutBucketEncryption",

"oss:DeleteBucketEncryption", "oss:RestoreObject", "oss:PutObjectTagging", "oss:GetObjectTagging", "oss:DeleteObjectTagging"

2.8.6. Configure an alert rule to monitor CSG gateways in the CloudMonitor console

This topic describes how to configure an alert rule to monitor Cloud Storage Gateway (CSG) gateways in the CloudMonitor console.

Prerequisites

- The endpoint of the CloudMonitor console is obtained from the deployment personnel before you log on to the CloudMonitor console.
- Google Chrome is used. We recommend that you use Google Chrome as the browser.

Log on to the CloudMonitor console

- 1. In the address bar of your browser, type the endpoint of the CloudMonitor console and press the Enter key.
- 2. Enter your username and password.
 - Obtain the username and password that you use to log on to the Apsara Uni-manager Management Console from the operations administrator.

() Note When you log on to the console for the first time, you must change the password as prompted. The password must be 8 to 20 characters in length and contain at least two of the following character types:

- Uppercase or lowercase letters
- Digits
- Special characters, including exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)
- 3. Click Log On.
- 4. In the top navigation bar, choose **Products > Monitoring and O&M > Cloud Monitor**.

Create an alert rule

- 1. On the Cloud Monitor page, click Cloud Service Monitoring in the left-side navigation pane.
- 2. On the Cloud Service Monitoring page, click CSG.
- 3. Click Create Alert Rule.
- 4. In the Create Alert Rule panel, configure the parameters and click OK. The following table describes the parameters.

Parameter	Description			
Product	Select Cloud storage gateway.			
Resource Range	Select the gateway to be monitored.			
Rule Description	Specify the description of the alert rule.			
Add Rule Description	Click Add Rule Description to create a rule. For more information, see Add Rule Description.			
Mute For	Select a proper mute period.			
Effective Period	Specify a time period during which the alert rule is effective.			
Alert Callback	Enter a URL for alert callback. The callback URL must start with http:// and cannot be an IP address.			
Alert Contact Group	Select an alert contact group.			

Table 1. Add Rule Description

Parameter	Description			
Rule Name	Enter an alert rule name.			
Metric Name	 Select the name of the metric that you want to monitor. Valid values: Gateway CPU usage percentage Gateway memory usage percentage Gateway cache usage percentage Gateway running state (We recommend that you create an alert rule for this metric.) Gateway file share throttling state (We recommend that you create an alert rule for this metric.) Gateway file share throttling state (We recommend that you create an alert rule for this metric.) Gateway file share throttling state (We recommend that you create an alert rule for this metric.) Gateway shared read/write rate Gateway file share read iops Gateway file share write iops Gateway file share upload OSS rate 			
Comparison	 Specify a rule to compare the metric value with the threshold. When the comparison rule is satisfied, an alert is triggered. >=: An alert is triggered when the metric value is greater than or equal to the threshold. >: An alert is triggered when the metric value is greater than the threshold. <=: An alert is triggered when the metric value is less than or equal to the threshold. <: An alert is triggered when the metric value is less than or equal to the threshold. <: An alert is triggered when the metric value is less than or equal to the threshold. <: An alert is triggered when the metric value is less than the threshold. =: An alert is triggered when the metric value is equal to the threshold. I=: An alert is triggered when the metric value is not equal to the threshold. Compared With Yesterday Rise: An alert is triggered when the metric value is greater than the value at the same time yesterday. Compared With Yesterday Decline: An alert is triggered when the metric value is less than the value at the same time yesterday. 			
Threshold and Alert Level	Configure the thresholds and alert levels.			

On the Cloud Service Monitoring page, click CSG. Then, click Alert Rules in the Actions column of a gateway to view the alert rules.

View monitoring charts

You can perform the following steps to view the details of metrics in the monitoring charts on the CSG page.

1. On the Cloud Service Monitoring page, click CSG.

2. Click Monitoring Charts in the Actions column of a gateway.

Recommended configurations and handling methods for some alert rules

Metric	Comparison operator	Threshold	Description
Gateway running state	==	0	If an alert is triggered for this metric, the gateway is not running properly. In this case, you can log on to the console to restart the gateway. To restart the gateway, choose MoreRestart the gateway in the Actions column.
Gateway file share remaining meta space	<	3221225472	If an alert is triggered for this metric, the metadata of the gateway is less than 3,221,225,472 bytes (3 GB). We recommend that you scale up the cache disk used by the share. For more information, see Scale up a cache disk.
Gateway file share throttling state		1	 If an alert is triggered for this metric, throttling has been enabled for the share. Check whether a file whose size exceeds the size of the cache disk used by the share is being written. If such a file exists, we recommend that you cancel the writing and then scale up the cache disk. For more information, see Scale up a cache disk. If no such a file exists, contact the system administrator.

3.EBS 3.1. EBS

Elastic Block Storage (EBS) is a persistent random block storage service with low latency and high reliability and is designed for Elastic Compute Service (ECS).

Overview

EBS provides EBS devices based on a distributed storage architecture. The EBS console provides you with an all-in-one solution for EBS management in the cloud. You can use the EBS console to manage enterprise-level features and other features such as storage resource management, monitoring, performance analysis, disaster recovery, and alerting. As a centralized service platform for the entire EBS infrastructure, the EBS console informs you of resources in the cloud to best support business and optimize costs by analyzing long-term data trends.

Terms

block storage

Block storage devices offer high performance and reduce latency. You can partition and format these devices and create file systems on the devices to meet the data storage requirements of your business.

EBS

EBS provide ECS instances with block-level storage that features low latency, persistent, and high-reliability. EBS devices use the triplicate distributed mechanism to ensure data durability for ECS instances. EBS devices can be created, released, and resized at any time.

() Note EBS devices including system and data disks can be resized online without interrupting their services. When you are resizing an EBS device, you do not need to stop the ECS instance to which the EBS device is attached or detach the EBS device from the ECS instance.

disk

Disks are block-level EBS devices that use the triplicate mechanism and support erasure coding (EC).

snapshot

A snapshot is a point-in-time backup of a disk and is used to back up or restore the disk.

async replication

The async replication feature can asynchronously replicate data from a primary disk in one zone to a secondary disk in another zone in the same region on a periodic basis. In async replication, the data on the primary disk may be inconsistent with that on the secondary disk.

EBS

EBS provides disks. A disk can be attached to a single ECS instance that resides within the same zone as the disk.

Disks

Disks are block-level storage devices designed for ECS instances. Disks are classified into the following categories based on their performance or purposes.

Performance-based classification

Disks are classified by performance into premium performance disks, standard performance disks, ultra disks, and standard SSDs.

- () Important
 - If you deploy an EBS cluster of the latest version, you can use only premium performance disks and standard performance disks.
 - If you update an EBS cluster from an earlier version to the latest version, you can continue to use ultra disks and standard SSDs in addition to newly supported premium performance disks and standard performance disks.
- Standard performance disks and premium performance disks are ideal for online transaction processing (OLTP) databases and NoSQL databases. Premium performance disks deliver up to 25,000 random IOPS for ECS instances.
- Ultra disks are ideal for medium I/O load scenarios and deliver up to 3,000 random IOPS for ECS instances.
- Standard SSDs are ideal for I/O-intensive applications and deliver stable and high random IOPS performance.

The following table compares the performance of different disks.

Comparison item	Premium performance disk	Standard performance disk	Standard SSD	Ultra disk
Maximum capacity per disk (GiB)	32,768	32,768	32,768	32,768
Maximum IOPS	25,000	5,000	25,000	5,000
Maximum throughput (MB/s)	300	140	300	140
Formula for calculating the IOPS per disk	min(1,800 + 30 × Capacity, 25,000)	min(1,800 + 8 × Capacity, 5,000)	min(1,800 + 30 × Capacity, 25,000)	min(1,800 + 8 × Capacity, 5,000)
Formula for calculating the throughput per disk (MB/s)	min(120 + 0.5 × Capacity, 300)	min(100 + 0.15 × Capacity, 140)	min(120 + 0.5 × Capacity, 300)	min(100 + 0.15 × Capacity, 140)
API parameter value	cloud_pperf	cloud_sperf	cloud_ssd	cloud_efficiency
Scenario	 OLTP databases: relational databases such as MySQL, PostgreSQL, Oracle, and SQL Server databases NoSQL databases: non-relational databases such as MongoDB, HBase, and Cassandra databases Elasticsearch distributed logs: Elasticsearch, Logstash, and Kibana (ELK) log analysis 		Small and medium-sized development and test environments that require high data reliability	 Development and test applications System disks

Purpose-based classification

Disks are classified into system disks and data disks based on their purposes.

- System disks are created and released along with the ECS instances to which they are attached and have the same lifecycle as the instances. Shared access is not allowed for system disks.
- Data disks can be created separately or along with ECS instances. Shared access is not allowed for data disks. A data disk created together with an ECS instance has the same lifecycle as the instance, and is released along with the instance. Data disks that are separately created can be released along with or independent of the ECS instance to which they are attached. The maximum capacity that a data disk can have is determined by its category.

Shared disks

⑦ Note Shared disks are being phased out.

Shared disks are a block-level data storage service that supports concurrent read and write operations on multiple ECS instances and offers high performance and high reliability.

A single shared disk can be attached to a maximum of four ECS instances. Shared disks can only be used as data disks and must be created separately. Shared access is allowed. You can configure shared disks to release along with the instances to which the disks are attached.

- Shared disks can be classified into the following types based on performance:
- Shared standard SSD: uses SSDs as the storage medium to provide stable and high performance storage that offers enhanced random I/O and data reliability.
- Shared ultra disk: uses a hybrid SSD and HDD storage medium.
- An ECS instance can have up to 16 data disks including cloud disks and shared disks.

3.1.1. Features

3.1.1.1. Overview

Elastic Block Storage (EBS) provides various features such as read/write stability, data encryption, backup, elastic storage, and disaster recovery.

Read/write stability

Three copies of your business data are stored in an EBS cluster within the same zone to ensure data stability during read and write operations. For more information, see Triplicate storage.

By default, EBS clusters use triplicate storage to ensure data reliability and consistency. Erasure coding (EC) can improve storage reliability. Compared to triplicate storage, EC can provide higher data reliability at lower data redundancy levels. You can enable EC for EBS clusters. For more information, see Erasure coding.

Backup

To improve the security of your business data, you can create snapshots on a periodic basis to provide data backup capabilities for EBS devices. This ensures that information such as logs and customer transactions is backed up for future use. For more information, see Overview.

You can use automatic snapshot policies to improve data security and tolerance against operation faults. Automatic snapshot policies can be applied to system disks and data disks to create periodical snapshots of the disks.

To simultaneously create snapshots for multiple EBS devices on an Elastic Compute Service (ECS) instance, you can create a snapshot-consistent group by adding the EBS devices. If the data of your business system is stored on multiple EBS devices, you can add the EBS devices to a snapshot-consistent group to ensure point-in-time consistency of data writes to the EBS devices and crash consistency of the data.

Data encryption

If your applications are data-sensitive, we recommend that you encrypt the EBS devices that you use. EBS devices and their snapshots are encrypted by using keys based on the industry-standard AES-256 algorithm. Data is automatically encrypted when it is transmitted from ECS instances to EBS devices. Encrypted data is automatically decrypted when it is read. For more information, see EBS device encryption.

Elastic storage

You can resize EBS devices to meet increasing storage requirements as your business and application data grow. For more information, see EBS device resizing.

Disaster recovery

Async replication is a feature that protects data across zones in a region based on the data replication capability of EBS. This feature can asynchronously replicate data between EBS devices in different zones in the same region for disaster recovery. You can use this feature to implement disaster recovery for critical business to protect data in your databases and improve business continuity. For more information, see Async replication.

If a replication pair and a replication pair-consistent group have the same primary region (production region), primary zone (production zone), secondary region (disaster recovery region), and secondary zone (disaster recovery zone), the replication pair and replication pair-consistent group replicate data in the same direction. You can add replication pairs that replicate data between the production site and disaster recovery site in the same direction as a replication pair-consistent group to the group so that you can manage these replication pairs a batch by using the group.

3.1.1.2. Triplicate storage

Apsara Distributed File System provides stable, efficient, and reliable data access to ECS instances.

Chunks

When ECS users perform read and write operations on virtual disks, the operations are translated into the corresponding processes on the files stored in Apsara Stack data storage system. Apsara Stack uses a flat design in which a linear address space is divided into slices called chunks. Each chunk is replicated into three copies. Each copy is stored on a different node in the cluster, which ensures data reliability.

Figure 1. Triplicate backup



How triplicate technology works

Triplicate storage is made up of three components: master, chunk server, and client. Each write operation performed by an ECS user is converted into an operation executed by the client. The execution process is as follows:

- 1. The client determines the location of a chunk corresponding to the write operation.
- 2. The client sends a request to the master to query the chunk servers where the three chunk replicas are each stored.
- 3. The client sends write requests to the chunk servers based on the results returned from the master.
- 4. If the three replicas of the chunk are all successfully written as requested, the client returns a message to indicate the success of the operation. If the write operation fails, a failure message is returned.

The master component distributes chunks based on the disk usage, rack distribution, power supply, and machine workloads of chunk servers. This ensure that chunk replicas are each distributed to chunk servers on different racks and that data does not become unavailable due to the failure of a single server or rack.

Data protection mechanism

When a data node is damaged or disk faults occur on a data node, the total number of valid replicas of some chunks in a cluster becomes less than three. In these cases, the master replicates data between chunk servers to ensure that there are always three valid replicas of chunks in the cluster.

Figure 2. Automatic replication



All user-level operations for data on cloud disks are synchronized across the three chunk replicas at the underlying layer. Operations that are synchronized include adding, modifying, and deleting data. This mode ensures the reliability and consistency of user data.

To prevent data losses caused by viruses, accidental deletion, or malicious attacks, we recommend that you use other protection methods such as backing up data and taking snapshots in addition to triplicate storage. Implement all appropriate measures to ensure the security and availability of your data.

3.1.1.3. Erasure coding

Erasure coding (EC) can improve storage reliability. Compared to triplicate storage, EC can provide higher data reliability at lower data redundancy levels.

What is EC?

- EC involves the following concepts:
- Data fragments (m): Data is divided into m data fragments.
- Parity fragments (n): n parity fragments are computed from the m data fragments.

The m data fragments and n parity fragments compose an erasure coding group. The data fragments and parity fragments are located on different servers. When n or less than n segments are lost, the lost segments can be restored based on the erasure coding algorithm. Both m and n are configurable. The typical configuration for Apsara Stack is 8 + 3, with the number of servers being no less than 14.

Comparison between EC and triplicate storage

Compared to triplicate storage, EC is a better solution in terms of storage usage and data reliability.

Item	EC	Triplicate storage
Storage usage	$m/\left(m+n\right)$: When m is 8 and n is 3, the storage usage is calculated based on the following formula: 8/(8 + 3) = 72.7%.	1/3 = 33.3%
Reliability	Allows up to n fragments to be lost. Failures on up to n servers are allowed in the worst case. For example, when m is 8 and n is 3, failures on up to three servers are allowed.	Allows up to two replicas to be lost. Failures on up to two servers are allowed in the worst case.

3.1.1.4. EBS device encryption

Elastic Block Storage (EBS) device encryption is a simple and secure method that can be used to encrypt the EBS devices that you create.

EBS device encryption eliminates the need to create or maintain your own key management infrastructure, change existing applications and O&M processes, or perform additional encryption operations. EBS device encryption does not have negative impacts on your business. You can use EBS device encryption to encrypt the following types of data:

Data stored on EBS devices.

• Data transmitted from Elastic Compute Service (ECS) instances to EBS devices. Data within the instance operating system is not encrypted.

All snapshots created from encrypted EBS devices. These snapshots are encrypted snapshots.

Data transmitted from ECS instances to EBS devices is encrypted on the hosts on which the ECS instances are deployed.

All available cloud disks, including premium performance disks, standard performance disks, ultra disks, and standard SSDs, and Shared Block Storage devices, including ultra Shared Block Storage devices and SSD Shared Block Storage devices, in Apsara Stack ECS can be encrypted.

3.1.1.5. EBS device resizing

You can resize EBS devices to meet increasing storage requirements as your business and application data grow. This topic describes how to resize EBS devices and provides usage notes for resizing EBS devices.

Scenarios

You can use one of the following methods to increase the storage capacity of an Elastic Compute Service (ECS) instance:

• Resize an existing EBS device. You can resize the existing partitions of the EBS device or create more partitions for the EBS device. The following table describes the two methods for resizing an existing EBS device.

Method	Usage notes	References
Resize an existing EBS device online	The ECS instance to which the EBS device is attached must be in the Running state. After you resize the EBS device, the new size takes effect without requiring you to restart the ECS instance.	For more information, see the <i>Resize disks</i> topic under EBS of CDS User Guide .
Resize an existing EBS device offline	The ECS instance to which the EBS device is attached must be in the Running or Stopped state. After you resize the EBS device, you must restart the ECS instance by using the ECS console or by calling the RebootInstance operation for the new size to take effect.	For more information, see the <i>Resize disks</i> topic under EBS of CDS User Guide .

• Create an EBS device, attach the EBS device to the ECS instance, and then partition and format the EBS device.

• Replace the system disk of the ECS instance and specify a greater size for the new system disk.

Maximum size of a resized system disk

The new size of a resized system disk must be greater than the original size but less than or equal to 500 GiB. The following table describes the maximum sizes of resized system disks that correspond to different images used by ECS instances.

Image	Maximum size of a resized system disk (GiB)
CoreOS and FreeBSD	[Max(30, Original size of the system disk), 500]
Other Linux distributions	[Max(20, Original size of the system disk), 500]
Windows Server	[Max(40, Original size of the system disk), 500]

For example, the system disk of a CentOS-based ECS instance is 35 GiB in size. When you resize the system disk, the specified new size must be greater than 35 GiB but less than or equal to 500 GiB.

Maximum size of a resized data disk

The new size of a resized data disk must be greater than the original size. Premium performance disks, standard performance disks, ultra disks, standard SSDs, ultra Shared Block Storage devices, and SSD Shared Block Storage devices can be resized to up to 32,768 GiB.

3.1.1.6. Snapshots

3.1.1.6.1. Overview

A snapshot is a copy of data on a cloud disk at the point in time that the snapshot is created.

You can use snapshots in scenarios such as environment replication and disaster recovery:

- You may want to use the data of one disk as the basis to write or store data to a different disk. To achieve this, you can create a snapshot for a cloud disk and then create another cloud disk from the snapshot. The new disk contains the basic data of the original disk.
- While cloud disks are a secure way to store data, their data may be subject to errors caused by application errors or malicious read and write operations and requires additional safeguard mechanisms. You can create snapshots at regular intervals to restore data to a previous point in time in case of data errors.

3.1.1.6.2. Mechanisms

This topic describes snapshots. Snapshots retain a copy of data stored on a disk at a certain point in time. You can schedule disk snapshots to be created periodically to ensure continuous operation of your business.

Snapshots are created incrementally such that only data changes between two snapshots are copied instead of all of the data, as shown in Snapshots.

Figure 1. Snapshots



Snapshot 1, Snapshot 2, and Snapshot 3 are the first, second, and third snapshots of a disk. When a snapshot is created, the file system checks each block of data stored on the disk, and only copies the blocks of data that differ from those on the previous snapshots. The changes between snapshots in the preceding figure are described as follows:

- All data on the disk is copied to Snapshot 1 because it is the first disk snapshot.
- The changed blocks B1 and C1 are copied to Snapshot 2. Blocks A and D are referenced from Snapshot 1.
- The changed block B2 is copied to Snapshot 3. Blocks A and D are referenced from Snapshot 1, and block C1 is referenced from Snapshot 2.
- When the disk needs to be restored to the status of Snapshot 3, snapshot rollback will copy blocks A, B2, C1, and D to the disk, which will be restored to the status at the time of Snapshot 3.
- If Snapshot 2 is deleted, block B1 in the snapshot is deleted, but block C1 is retained because it is referenced by other snapshots. When you roll back a disk to Snapshot 3, block C1 is recovered.

③ Note Snapshots are stored on the Object Storage Service (OSS), but are hidden from users. Snapshots do not consume bucket space in OSS. Snapshot operations can only be performed from the ECS console or through APIs.

3.1.1.6.3. Specifications of ECS Snapshot 2.0

Built on the features of the original snapshot service, the ECS Snapshot 2.0 data backup service provides a higher snapshot quota and a more flexible automatic snapshot policy. This service has less impact on business I/O.

Table 1. Comparison of snapshot specifications

Item	Traditional snapshot specification	Snapshot 2.0 specification	Benefit	Example
Snapshot quota	Maximum allowable number of snapshots: Number of disks × 6 + 6.	Each disk can have up to 64 snapshots.	Longer protection cycle and smaller protection granularity.	 A snapshot is created for the data disks of non-core business at 00:00 every day. Snapshots taken within the last two months are retained. A snapshot is created for the data disks of core business every four hours. Snapshots taken within the last ten days are retained.
Automatic snapshot policy	By default, the task is scheduled to be triggered once a day and cannot be modified manually.	You can customize the time of day and days of the week that snapshots are scheduled to be created and the retention period of snapshots. The disk quantity and related details associated with an automatic snapshot policy can be queried.	More flexible protection policy.	 You can schedule snapshots to be created on the hour several times in a single day. You can specify the days of the week for which to create snapshots. You can specify the snapshot retention period or choose to retain a snapshot permanently. When the number of automatic snapshots reaches the upper limit, the oldest automatic snapshot will be automatically deleted.
Implementation	Copy-on-write (COW)	Redirect-on-write (ROW)	Mitigates the impact of snapshot tasks on business I/O performance.	Snapshots can be taken at any time without interruptions to your business.

3.1.1.7. Async replication

Async replication is a feature that protects data across zones within the same region based on the data replication capability of Elastic Block Storage (EBS). This feature can asynchronously replicate data between disks across zones within the same region for disaster recovery. You can use this feature to implement disaster recovery for critical business to protect data in your databases and improve business continuity.

Introduction

The async replication feature can asynchronously replicate data from a primary disk in a zone to a secondary disk that has the same specifications as the primary disk in another zone of the same region. If the primary disk fails, you can fail over to the secondary disk. After the primary disk recovers, you can restore data from the secondary disk.

The following table describes the operations related to async replication.

Feature	Operation	Description
	Create a replication pair	Before you can use the async replication feature to implement disaster recovery across zones in the same region, you must create a replication pair.
	Enable async replication	After you create a replication pair by specifying a primary disk and a secondary disk, you must enable the async replication feature to replicate data from the primary disk to the secondary disk across zones in the same region on a periodic basis.

Cloud Defined Storage

User Guide-EBS

Async replication	Disable async replication	After you enable the async replication feature, you can disable the async replication feature if you no longer require data replication or you want to perform a failover.
	Implement disaster recovery	After you create and activate a replication pair, if the primary disk fails, you can perform a failover and reverse replication to implement disaster recovery.
	Delete a replication pair	Specific limits are imposed on the disks for which you create a replication pair. Therefore, if a replication pair is no longer needed, you can delete it. After the replication pair is deleted, the secondary disk rolls back to the point in time when the last async replication was complete and drops all the data that is being replicated from the primary disk.
		You can create a replication pair-consistent group to operate and manage replication pairs between the primary site and the secondary site in a centralized manner.
Replication pair- consistent group	Create a replication pair- consistent group	A replication pair and a replication pair-consistent group replicate data in the same direction if they have the same primary region (production region), primary zone (production zone), secondary region (disaster recovery region), and secondary zone (disaster recovery zone). A replication pair can be added to a replication pair- consistent group only when they replicate data in the same direction.
	Add replication pairs	You can add replication pairs that replicate data in the same direction as a replication pair-consistent group to the group so that you can manage these replication pairs at a time by using the group.
	Remove replication pairs	You can remove multiple replication pairs from a replication pair-consistent group at a time. When a replication pair is removed from a replication pair-consistent group, the replication pair is disassociated from the group but is not deleted.
	Enable async replication for a replication pair- consistent group	After you add replication pairs to a replication pair-consistent group, you must activate the replication pairs by enabling async replication for the group to asynchronously replicate data from disks in the primary site to disks in the secondary site on a periodic basis. After async replication is enabled for a replication pair-consistent group, the system first performs a full synchronization to synchronize all data from disks in the primary site to disks in the secondary site. Then, the system periodically synchronizes incremental data based on the recovery point objective (RPO) of the replication pair-consistent group.
	Disable async replication for a replication pair- consistent group	After async replication is enabled for a replication pair-consistent group, if you no longer require data replication or you want to perform a failover, you can disable the async replication feature for the replication pair-consistent group.
	Implement disaster recovery	After you create and activate a replication pair-consistent group, if disks in the primary site fail, you can use the async replication feature to implement disaster recovery for the primary disks.
	Delete a replication pair- consistent group	You can delete a replication pair-consistent group that you no longer need. After the replication pair-consistent group is deleted, disks in the secondary site roll back to the point in time when the last async replication was complete and drop all the data that is being replicated from disks in the primary site.

The following table describes the terms related to the async replication feature.

Term	Description
async replication	The async replication feature can asynchronously replicate data from a primary disk in one zone to a secondary disk in another zone in the same region on a periodic basis. In async replication, the data on the primary disk may be inconsistent with that on the secondary disk.
primary disk	The disk from which to replicate data for disaster recovery.
secondary disk	The disk on which the replicated data is stored.
RPO	The amount of data that may be lost due to a disk exception. RPO is measured by time and is used as a metric for async replication. You can specify the value of RPO. Valid values of RPO range from 2 to 10080. Unit: minutes. For example, the value of RPO is set to 2 minutes. In this case, if an exception occurs on a primary disk, incremental data written to the primary disk from up to 2 minutes prior to the exception may be lost when the primary disk recovers.
recovery time objective (RTO)	The duration of time that it takes for a primary disk to recover after an exception occurs on the disk. RTO is used as a metric in async replication. For example, if the value of RTO is 1 hour, a primary disk can have its data restored and be back in service within 1 hour after an exception occurs on the primary disk.
replication pair	The replication relationship that is established between a primary disk, a secondary disk, and configurations for asynchronous replication.
failover	A sub-feature of async replication that allows you to enable read and write permissions on the secondary disk and fail over to the secondary disk if the primary disk fails.
reverse replication	A sub-feature of async replication that can reverse replication relationships to replicate data from the secondary disk to the primary disk.

Use scenarios

The async replication feature is suitable for users who require high data security and need to implement disaster recovery and cross-zone migrations of business data.

Disaster recovery

If the primary disk in a replication pair fails, you can perform a failover to switch the primary and secondary disks over. During the failover, the failover feature disconnects the original replication link and fails services over to the disaster recovery system by attaching the new primary disk (original secondary disk) to an Elastic Compute Service (ECS) instance that is used for disaster recovery.

Cross-zone data migration in the same region

If you want to migrate your business data across zones within the same region, you can use the reverse replication sub-feature, instead of the image or snapshot replication feature.

Precautions

The following table describes the limits that apply to the async replication feature.

Item	Limits
Item	Limits

Replication pairs that can be created for a single disk	1
Replication cycle	Async replication is automatically performed based on the specified RPO period. Valid values of RPO: 2 to 10080. Unit: minutes.
Primary disk category	A primary disk must be a premium performance disk.
Secondary disk category	A secondary disk must be of the same disk type and have the same performance level and capacity as the associated primary disk.

The limits described in the following table apply to primary and secondary disks when you use the async replication feature.

Item	Method	Support for primary disks	Support for secondary disks	Description
Attach a disk	Use the ECS console	1	\checkmark	For a secondary disk, if you attach the disk to an ECS instance when the instance is running, the disk can be attached to the instance but no data can be read from or written to the disk. If you attach the disk when the ECS instance stops, the ECS instance fails to start next time.
Use a system disk	Use the ECS console		V	 If the system disk of an ECS instance is used as a primary disk, the ECS instance can read and write data from and to the disk as expected. If the system disk of an ECS instance is used as a secondary disk, the ECS instance may fail to start, or you may fail to access the operating system of the instance after startup.
Read and write data from and to disks	Use the ECS console	V	×	After a replication pair is activated, the secondary disk enters the read-only state, and no users have write permissions on the secondary disk.
Delete a disk	DeleteDisk	/	\checkmark	After a disk is deleted for a period of time, the state of the replication pair to which the disk belongs changes to Invalid .
lnitialize a disk	ReinitDisk	/	\checkmark	After a disk is initialized for a period of time, the state of the replication pair to which the disk belongs changes to Invalid .
Create a snapshot	Use the ECS consoleReinitDisk	1	v	Due to RPO, data of a snapshot created for a primary disk may not be consistent with that of a snapshot created at the same time for the associated secondary disk.
Roll back a disk by using its snapshot	Use the ECS consoleResetDisk	×	×	 If the disk is a primary disk, the disk can be rolled back by using its snapshot. If the disk is a secondary disk, the disk cannot be rolled back by using its snapshot.
Replace a system disk	Use the ECS consoleReplaceSystemDisk	,	V	After you perform this operation, the original disk is deleted. Therefore, this operation is equivalent to the DeleteDisk operation. After the system disk of an ECS instance is replaced for a period of time, the state of the replication pair to which the system disk belongs changes to Invalid .
Resize a disk online	ResizeDisk	1	v	After you resize a disk online, the state of the replication pair to which the disk belongs changes to Invalid .
Resize a disk offline	ResizeDisk		√	After you resize a disk offline, the state of the replication pair to which the disk belongs changes to Invalid . () Important After you resize a disk offline, make sure that the new disk size takes effect before you create a replication pair for the disk. Otherwise, the replication pair may become invalid.
Change the type of a disk	Not supported	×	x	N/A.
Encrypt a disk	Not supported	×	×	You cannot convert a non-encrypted disk to an encrypted disk.
Enable multi-attach	Not supported	×	×	You cannot enable the multi-attach feature for a disk.
Migrate disks along with ECS instances	Not supported	×	×	N/A.

3.2. Log on to the Apsara Uni-manager Management Console

This topic describes how to log on to the Apsara Uni-manager Management Console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.
- We recommend that you use the Google Chrome browser.

Procedure

1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.

2. Enter your username and password.

Obtain the username and password from an operations administrator.

? Note

• First logon

The first time that you log on to the Apsara Uni-manager Management Console, you need to change the password of your account. The password must be 10 to 32 characters in length. It must contain all of the following character types: uppercase letters, lowercase letters, digits, and special characters. Supported special characters include: ! @ #

• Forget password

If you have forgotten your password, click Forgot Password. On the page that appears, enter the username of your account, the email address that was used to create the account, and the CAPTCHA code. Then, the system sends a link for resetting the password to the specified email address.

3. Click Log On.

4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations in the following scenarios:

- $\circ~$ It is the first time that you log on to the console after MFA is forcibly enabled by the administrator.
- a. On the Bind Virtual MFA Device page, bind an MFA device.
- b. Enter the account and password again as in Step 2 and click Log On.
- c. Enter a six-digit MFA verification code and click **Authenticate**.
- You have enabled MFA and bound an MFA device.

Enter a six-digit MFA authentication code and click Authenticate.

② Note For more information, see the Bind a virtual MFA device to enable MFA topic in Apsara Uni-manager Operations Console User Guide.

3.3. Getting started

3.3.1. Create a disk

You can separately create a data disk and then attach it to an Elastic Compute Service (ECS) instance to increase the storage space of the instance. This topic describes how to create an empty data disk. You cannot separately create system disks in ECS.

Background information

We recommend that you determine the number and sizes of data disks before you create them. Take note of the following limits:

- Up to 16 data disks can be attached to an instance. Disks and Shared Block Storage devices share this quota.
- Each Shared Block Storage device can be attached to up to four instances at the same time.
- Each premium performance disk, standard performance disk, ultra disk, ultra Shared Block Storage device, standard SSD, or SSD Shared Block Storage device can have a maximum capacity of 32 TiB.
- Disks cannot be merged in ECS. Disks are independent of each other. You cannot merge the capacity of disks by formatting the disks.

We recommend that you do not use Logical Volume Manager (LVM) to create logical volumes across multiple disks because a snapshot can back up the data of only a single disk. If you create a logical volume across disks, data inconsistency may occur when you roll back a disk by using a snapshot.

Procedure

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Disks.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. In the upper-left corner of the **Disks** page, click **Create Disk**.
- Set the parameters as required and click Submit. The following table describes the parameters. After the disk is created, click OK in the message that appears to return to the Disks page.

Section	Parameter	Required	Description
	Organization	Yes	The organization to which the disk belongs.
	Resource Set	Yes	The resource set to which the disk belongs.
	Region	Yes	The region in which the disk is located.
	Zone	Yes	The zone in which the disk is located.
Area			

		Creation Method	Yes	 The method to create the disk. Valid values: Disk Creation and Storage Set Creation. If you want to create the disk in a partition of a specified storage set, select Storage Set Creation. Important If you can select only Disk Creation, no storage sets are available in the current environment. In this case, configure a storage set first. Before you create a disk in a partition of a storage set, make sure that the storage set is created and the partition is configured. For more information about how to create a storage set, seeCreate a storage set.
	Basic Configurations	Storage Set	Yes	The name of the storage set. This parameter is required only if you set the Creation Method parameter to Storage Set Creation.
		Partitions	Yes	The sequence number of the partition. This parameter is required only if you set the Creation Method parameter to Storage Set Creation .
		Name	Yes	The name of the disk. The name must be 2 to 128 characters in length. It must start with a letter and cannot start with http:// or https://. It can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,). If you create multiple disks at a time, the system automatically adds numbers such as 001 and 002 to the specified disk name. For example, if you set the Name parameter to mydisk and create two disks at a time, the names of the created disks are mydisk001 and mydisk002.
		Quantity	Yes	The number of disks to create. You can create multiple disks at a time. Valid values: 1 to 100. Default value: 1.
		Specifications	Yes	The disk type. After you select a disk type, specify a disk size. Valid values: Ultra Disk Premium Performance Disk Standard SSD Standard Performance Disk Shared SSD: SSD Shared Block Storage device Shared Ultra Disk: Ultra Shared Block device The specified disk size must range from 20 GiB to 32,768 GiB.
		Encryption	No	Specifies whether to encrypt the disk. By default, the disk is not encrypted.
		Use Snapshot	No	 Specifies whether to create the disk based on a snapshot. If you selectYes, you must specify a snapshot. The size of the created disk depends on the size of the specified snapshot. If the disk size that you specify is greater than the snapshot size, the disk is created with the size you specify. If the disk size that you specify is smaller than the snapshot size, the disk is created with the snapshot size.

Result

The created disk is displayed in the disk list and is in the Available state.

What to do next

You need to attach the disk to an instance. For more information, see Attach disks.

3.3.2. Attach disks

You can attach disks that are separately created as data disks to instances. Before you attach a disk to an ECS instance, make sure that the disk and the instance are located in the same region and zone.

Prerequisites

- The disks and the instance to which the disks are to be attached are located in the same zone.
- The instance is in the Running or Stopped state.
- The disks are in the Available state.

Background information

Before you attach disks, take note of the following items:

- Each data disk that is created along with an instance is automatically attached to the instance.
- A disk can be attached to only an instance that is in the same zone and region as the disk.
- Each disk can be attached to only one instance at the same time.
- Each Shared Block Storage device can be attached to up to four instances at the same time.

Attach disks on the instance details page

To attach multiple disks to an ECS instance, we recommend that you go to the details page of the instance.

1. Log on to the ECS console.

- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose **Instances & Images > Instances**.
- 3. In the top navigation bar, select an organization, a resource set, and a region.

- 4. Find the ECS instance to which you want to attach disks and click the instance ID.
- 5. Click the **Disks** tab. In the upper-left corner of the Disks tab, click **Attach Disk**.
- 6. In the Attach Disk dialog box, select the disks that you want to attach to the instance.
- 7. Click **OK**.

Attach disks on the Disks page

To attach multiple disks to different ECS instances, we recommend that you go to the Disks page.

1. Log on to the ECS console.

- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Disks.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find a disk that you want to attach to an ECS instance, move the pointer over the ... icon in the Actions column, and then select Attach.
- 5. In the Attach dialog box, specify the instance and set the release mode.
 - If you select Release Disk with Instance, the disk is released when the ECS instance to which the disk is attached is deleted.
 - If you do not select Release Disk with Instance, the disk is retained and enter the Available state when the ECS instance to which the disk is attached is deleted.
- 6. Click **OK**.

What to do next

After the disk is attached to the instance, you need to initialize the disk. For more information, see Format a data disk on a Linux instance and Format a data disk on a Windows ECS instance.

3.3.3. Partition and format a disk

3.3.3.1. Format a data disk on a Linux instance

After you separately create a data disk, the data disk is not partitioned or formatted. This topic describes how to partition and format a data disk on a Linux instance.

Prerequisites

The disk is attached to an ECS instance.

Procedure

- 1. Connect to the instance. For more information, see the Connect to instances section in ECS User Guide.
- Run the fdisk -I command to view all data disks attached to the instance. If /dev/vdb is not displayed in the command output, no data disk is attached to the ECS instance. Check whether the data disk that you want to manage is attached to the instance.

```
[root@iZ*******eZ ~]# fdisk -1
Disk /dev/vda: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00078f9c
```

```
Device Boot Start End Blocks Id System

/dev/vda1 * 1 522 41940992 83 Linux

Disk /dev/vdb: 21.5 GB, 21474836480 bytes

16 heads, 63 sectors/track, 41610 cylinders

Units = cylinders of 1008 * 512 = 516096 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x0000000
```

- 3. Partition the data disk.
 - i. Run the **fdisk /dev/vdb** command.
- ii. Enter n to create a partition.
- iii. Enter p to set the partition type to primary partition.
- iv. Enter the partition sequence number and press the ENTER key. In this example, 1 is entered to create Partition 1.
- v. Enter a sequence number for the first available sector. You can press the ENTER key to accept the default value, or enter a value ranging from 1 to 41,610 and then press the ENTER key. In this example, the default value is used.
- vi. Enter the sequence number for the last sector. You can press the ENTER key to accept the default value, or enter a value ranging from 1 to 11,748 and then press the ENTER key. In this example, the default value is used.
- vii. Optional: (Optional) To create multiple partitions, repeat steps ii to vi until all four primary partitions are created.

viii. Run the wg command to start partitioning the data disk.

[root@iZ******eZ ~]# fdisk /dev/vdb Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel Building a new DOS disklabel with disk identifier 0x01ac58fe. Changes will remain in memory only, until you decide to write them. After that, of course, the previous content won't be recoverable. Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite) WARNING: DOS-compatible mode is deprecated. It's strongly recommended to switch off the mode (command 'c') and change display units to sectors (command 'u'). Command (m for help): n Command action e extended p primary partition (1-4) p Partition number (1-4): 1 First cylinder (1-41610, default 1): Using default value 1 Last cylinder, +cylinders or +size{K,M,G} (1-41610, default 41610): Using default value 41610 Command (m for help): wq The partition table has been altered! 4. Run the fdisk -I command to view the partitions. If /dev/vdb1 is displayed in the command output, a partition named vdb1 is created. [root@iZ******eZ ~]# fdisk -1 Disk /dev/vda: 42.9 GB, 42949672960 bytes 255 heads, 63 sectors/track, 5221 cylinders Units = cylinders of 16065 * 512 = 8225280 bytes Sector size (logical/physical): 512 bytes / 512 bytes I/O size (minimum/optimal): 512 bytes / 512 bytes Disk identifier: 0x00078f9c /dev/vdal * End End End Blocks Id System 5222 41940992 83 Linux Disk /dev/vdb: 21.5 GB, 21474836480 bytes 16 heads, 63 sectors/track, 41610 cylinders Units = cylinders of 1008 * 512 = 516096 bytes Sector size (logical/physical): 512 bytes / 512 bytes I/O size (minimum/optimal): 512 bytes / 512 byte Disk identifier: 0x01ac58fe t End Blocks Id System 1 41610 20971408+ 83 Linux Device Boot Start /dev/vdb1 5. Format the new partition. In this example, the mkfs.ext3 /dev/vdb1 command is used to format the new partition as ext3. The amount of time required for formatting depends on the disk size. You can also format the new partition to another file system. For example, run the **mkfs.ext4** /dev/vdb1 command to format the partition as ext4. Compared with ext2, ext3 only adds the logging feature. Compared with ext3, ext4 improves some important data structures. ext4 provides better performance and reliability, and more functions. [root@iZ******leZ ~] # mkfs.ext3 /dev/vdb1 mke2fs 1.41.12 (17-May-2010) Filesystem label= OS type: Linux Block size=4096 (log=2) Fragment size=4096 (log=2) Stride=0 blocks, Stripe width=0 blocks 1310720 inodes, 5242852 blocks 262142 blocks (5.00%) reserved for the super user First data block=0 Maximum filesystem blocks=4294967296 160 block groups 32768 blocks per group, 32768 fragments per group 8192 inodes per group Superblock backups stored on blocks: 32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208, 4096000

Writing inode tables: done

Creating journal (32768 blocks): done Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 25 mounts or 180 days, whichever comes first. Use tune2fs -c or -i to override.

6. Run the echo '/dev/vdb1 /mnt ext3 defaults 0 0' >> /etc/fstab command to write the information of the new partition to the /etc/fstab file. Then, run the cat /etc/fstab command to view the information of the new partition.

Ubuntu 12.04 does not support barriers. To write the information of the new partition to the /etc/fstab file, run the echo '/dev/vdb1 /mnt ext3 barrier=0 0 0' >> /etc/fstab command.

In this example, the partition information is added to the ext3 file system. You can also modify the ext3 parameter to add the partition information to another type of file system.

To mount the data disk to a specific folder for specific purposes, such as storing web pages, modify the /mnt part of the preceding command.

[root@iZ*******eZ ~]# echo '/dev/vdb1 /mnt ext3 defaults 0 0' >> /etc/fstab [root@iZbp19cdhgdj0aw5r2izleZ ~]# cat /etc/fstab

```
# /etc/fstab
# Created by anaconda on Thu Aug 14 21:16:42 2014
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
UUID=94e4e384-0ace-437f-bc96-057dd64f**** / ext4 defaults,barrier=0 1 1
                                                                    0 0
tmpfs
                      /dev/shm
                                             tmpfs defaults
                                             devpts gid=5,mode=620 0 0
devpts
                      /dev/pts
                                             sysfs defaults
                      /sys
sysfs
                                                                    0 0
                                                    defaults
                                                                    0 0
proc
                      /proc
                                            proc
/dev/vdb1 /mnt ext3 defaults 0 0
```

7. Run the following command to mount the new partition: Run the mount -a command to mount all the partitions listed in the /etc/fstab file and run the df -h command to view the result. If the following information is displayed, the new partitions are mounted and ready for use.

[root@iZ******	*eZ ~];	# mour	nt -a		
[root@iZ******	*eZ ~];	# df -	-h		
Filesystem	Size	Used	Avail	Use%	Mounted or
/dev/vda1	40G	5.6G	32G	15%	1
tmpfs	499M	0	499M	0%	/dev/shm
/dev/vdb1	20G	173M	19G	1%	/mnt

3.3.3.2. Format a data disk on a Windows ECS instance

After you separately create a data disk, the data disk is not partitioned or formatted. This topic uses Windows Server 2008 as an example to describe how to partition and format a data disk on a Windows instance.

Prerequisites

The disk is attached to an ECS instance.

Procedure

- 1. In the lower-left corner of the screen, click the Server Manager icon.
- 2. In the left-side navigation pane of the Server Manager window, choose Storage > Disk Management.
- 3. Right-click an empty partition and select New Simple Volume.
- If the data disk is in the **Offline** state, change it to the **Online** state.
- 4. Click Next.
- Specify the size of the simple volume, which is the partition size. Then, click Next. The default value is the maximum value of the disk space. You can specify the partition size based on your business requirements.
- 6. Specify the drive letter and click Next.
- 7. Configure the formatting options and click Next.
- We recommend that you format the partitions with the default settings provided by the wizard.
- 8. After you complete the configurations, click **Finish** to close the wizard.

3.4. Disks

3.4.1. Search for and view a disk

You can view all the disks that you create, search for a specific disk and view its details.

Procedure

- 1. Log on to the ECS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Disks.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. On the **Disks** page, use one of the following methods to search for the required disk:
- Select a field from the field drop-down list and enter a keyword in the search box. Then, click the Q icon. The disks that meet the filter condition are displayed in the list
- Click Advanced Filter. Specify values for different fields as required and click Search. The disks that meet the filter conditions are displayed in the list.

(?) Note When you use the advanced filtering feature, you can specify multiple filter conditions to narrow down search results.		
Field	Description	
Disk Name	Enter the name of a disk to search for the disk.	
Disk ID	Enter the ID of a disk to search for the disk.	
Encryption Key ID	Enter the ID of an encryption key to search for the disks that are encrypted by using the key.	

Instance ID	Enter the ID of an ECS instance to search for the disks that are attached to the instance.
Storage Set	Enter the ID of a storage set to search for the disks that belong to the storage set.
Partitions	Enter the sequence number of a partition to search for the disks that belong to the partition.
Snapshot Policy ID	Enter the ID of an automatic snapshot policy to search for the disks for which the policy is enabled.
Tag	Select a tag key or a tag value to search for the disks to which the tag is added. If you want to use tags to filter disks, make sure that you have added tags to your disks.

- 5. In the Disk ID/Name column, click the disk ID.
 - The basic information and attachment information are displayed in the disk details panel. You can perform the following operations based on your business requirements:
 - Modify the description of the disk
 - In the **Basic Information** section, click **Modify Disk Description** to modify the disk description. For more information, see Modify the name and description of a disk.
- Modify the properties of the disk
 - In the **Attachment Information** section, click **Modify Disk Properties** to modify the disk properties. For more information, see Modify the properties of a disk.

3.4.2. Modify the properties of a disk

You can modify the properties of a created disk, including Release Disk with Instance and Release Automatic Snapshots with Disk.

Procedure

- Log on to the ECS console.
 - i. Log on to the Apsara Uni-manager Management Console.
 - ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Disks.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find the disk that you want to modify, move the pointer over the ... icon in the Actions column, and then select Modify Disk Properties.
- 5. In the Modify Disk Properties dialog box, modify the release mode.
 - Release Disk with Instance: If you select this option, the disk is released when the instance to which the disk is attached is deleted. If you do not select this option, the disk enters the Available state after the instance to which the disk is attached is deleted.
 - Release Automatic Snapshots with Disk: If you select this option, the automatic snapshots created for the disk are released when the disk is deleted. If you do not select this option, the automatic snapshots are retained after the disk is deleted.
- 6. Click **OK**.

3.4.3. Modify the name and description of a disk

You can modify the name and description of a created disk.

Procedure

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose **Storage & Snapshots > Disks**.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find the disk that you want to modify, move the pointer over the ____ icon in the Actions column, and then select Modify Disk Description.
- 5. In the Modify Disk Description dialog box, modify the name and description of the disk. The name of the disk must be 2 to 128 characters in length and start with a letter. It can contain letters, digits, periods (.), underscores (_), hyphens (-), and colons (:).

The description of the disk must be 2 to 256 characters in length and cannot start with http:// or https://.

6. Click **OK**.

3.4.4. Resize a disk

You can resize system disks and data disks online. After a disk is resized, you do not need to restart the instance to which the disk is attached for the new disk capacity to take effect.

Prerequisites

- A snapshot is created. To avoid data loss, we recommend that you create a snapshot to back up the disk data before you resize a disk. For more information, see Create a snapshot.
- No snapshot is being created for the disk.
- The following requirements are met:
 - If the disk is a system disk, the ECS instance to which the disk is attached must be in the Running state.
 - If the disk is a data disk, one of the following requirements is met:
 - The disk must be in the **Available** state.
 - If the disk is attached to an ECS instance, the ECS instance must be in the **Running** state.
 - $\circ~$ If the disk is a Shared Block Storage device, the device must be in the $\ensuremath{\textbf{Available}}$ state.

Background information

When you resize a disk, take note of the limits described in the following table.

Limit	Description
Disk category	 Premium performance disks and standard performance disks can be resized. Ultra disks and standard SSDs can be resized. SSD Shared Block Storage and ultra Shared Block Storage devices can be resized.
Operating system	The system disks on Windows Server 2003 instances cannot be resized.
Partitioning mode	If a data disk adopts the MBR partition format, you cannot resize the data disk to more than 2 TiB. If you want to resize a data disk to 2 TiB and the data disk adopts the MBR partition format, we recommend that you create and attach another data disk. Then, format a GPT partition and copy the data in the MBR partition to the GPT partition.
File system	For Windows instances, only disks that use the NTFS file system can be resized.
Maximum capacity	 Premium performance disks and standard performance disks: 32,768 GiB Ultra disk and standard SSDs: 32,768 GiB SSD Shared Block Storage devices and ultra Shared Block Storage devices: 32,768 GiB
Related operations	 When you resize a disk, only the capacity of the disk is expanded. The sizes of partitions and file systems do not change. You must manually re-allocate the storage space on a disk after the disk is expanded. You cannot shrink a resized disk by means such as rolling it back.

Procedure

1. Log on to the ECS console.

i. Log on to the Apsara Uni-manager Management Console.

ii. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.

- 2. In the left-side navigation pane, choose Storage & Snapshots > Disks
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find the disk that you want to resize, move the pointer over the ____ icon in the Actions column, and then select Resize Disk.
- 5. In the **Resize Disk** dialog box, select a resizing method and specify the new capacity. You can set the Resizing Method parameter to Offline Resizing or Online Resizing. If you use the online resizing method, you can resize disks without the need to restart instances. Premium performance disks, standard performance disks, ultra disks, and standard SSDs are supported. If you use the offline resizing method to resize a disk, you must restart the instance involved in the console or call the RebootInstance operation to make the new disk capacity take effect.

() Important The new capacity must be greater than the current capacity.

6. Click **OK**.

Result

When you resize a disk, only the capacity of the disk is expanded. The sizes of partitions and file systems do not change. You must manually re-allocate the storage space on a disk after the disk is expanded.

3.4.5. Enable the multi-attach feature for disks that support NVMe

3.4.5.1. Overview of disks that support NVMe

Non-Volatile Memory Express (NVMe) is a host controller interface protocol used to accelerate the transfer of data from non-volatile memory. Alibaba Cloud premium performance disks support NVMe. Each premium performance disk can be simultaneously attached to multiple Elastic Compute Service (ECS) instances that support NVMe for data sharing. This topic describes premium performance disks that support NVMe, limits on attaching this type of disks, and operations related to this type of disks.

Premium performance disks that support NVMe

Premium performance disks can be attached to multiple ECS instances. After premium performance disks are attached to multiple instances, the disks support concurrent read and write access from these ECS instances and provide high reliability, high concurrency, and high performance. Premium performance disks provide the multi-attach and I/O blocking features.

- After the multi-attach feature is enabled for a premium performance disk, the disk can be attached to up to 16 ECS instances at a time.
- You can run NVMe commands to manage the permissions of ECS instances on premium performance disks. For more information about NVMe commands, see NVM Express Base Specification.

The preceding features improve service availability without compromising data reliability. If a single point of failure (SPOF) occurs, you can use a premium performance disk to quickly schedule and restore data. Data sharing among multiple ECS instances greatly reduces storage costs and improves service flexibility. Premium performance disks are suitable for high-availability databases and distributed database clusters that each consist of one write node and multiple read-only nodes.

Premium performance disks can be attached to ECS instances that support NVMe. For example, after premium performance disks are attached to Linux instances based on NVMe, you can run the **Isblk** command to check the device names and partition names of the disks, as shown in the following figure.



Description of the command output:

- The device names of the premium performance disks are displayed in the /dev/nvmeXn1 format. Examples: /dev/nvmeOn1, /dev/nvme1n1, and /dev/nvme2n1.
- The partition names of the premium performance disks are displayed in the CDevice name of the disk>p<Partition number> format. Examples:
 /dev/nvme0n1p1, /dev/nvme1n1p1, and /dev/nvme1n1p2.

Shared NVMe disks support the multi-attach feature. You can attach a shared NVMe disk to multiple ECS instances to help migrate high-availability services to the cloud. For more information, see Enable the multi-attach feature for disks.

Limits

Before you attach premium performance disks to an ECS instance based on NVMe, the resources of the instance must meet the limits described in the

following table.

Item	Limits
Instance family	By default, the instance family must support NVMe. The following instance families support NVMe: • ecs.ebmg7s-se-x25-c1m8 • ecs.ebmg7m-se-x25-c1m8 • ecs.ebmg7s-se-numaoff-x25-c1m8 • ecs.ebmg7m-se-numaoff-x25-c1m8 • ecs.ebmg7x-se-numaoff-x25-c1m8 • ecs.ebmg7x-se-numaoff-x25-c1m8 • ecs.ebmg7x-se-x25 ⑦ Note You can call the DescribeInstanceTypes operation to query instance families and check whether the instance family supports NVMe based on the NvmeSupport parameter in the response. For more information, see the DescribeInstanceTypes operation to query instance families and check whether the instance family supports NVMe based on the NvmeSupport parameter in the response. For more information, see the DescribeInstanceTypes operation to query instance families and check whether the instance family supports NVMe based on the NvmeSupport parameter in the response. For more information, see the DescribeInstanceTypes operation to query instance families and check whether the instance family supports NVMe based on the NvmeSupport parameter in the response. For more information, see the DescribeInstanceTypes operation to query instance families and check whether the instance family supports NVMe based on the NvmeSupport parameter in the response. For more information, see the DescribeInstanceTypes topic in ECS Developer Guide.
Image	The image must contain the NVMe driver. The NVMe driver is installed in the following public images. ③ Note Only some public Linux images support the NVMe driver. • CentOS 7: CentOS 7.6 and later • CentOS 8: CentOS 8.0 and later
Disk	 Disk category: premium performance disk Creation method: Create premium performance disks when you create instances that support NVMe. When you create disks, select premium performance disks as the disk category and enable the multi-attach feature.

3.4.5.2. Enable the multi-attach feature for disks

When you create a premium performance disk, you can enable the multi-attach feature for the disk. After multi-attach is enabled for a premium performance disk, the disk can be attached to up to 16 Elastic Compute Service (ECS) instances that support the Non-Volatile Memory Express (NVMe) protocol within the same zone to allow concurrent read and write access from the instances.

Benefits

This feature is suitable for high-availability databases and distributed database clusters that each consists of one write node and multiple read-only nodes. This feature provides the following benefits:

- Usage of NVMe commands: NVMe commands can be used to manage permissions of ECS instances on premium performance disks. This helps
 improve service availability without compromising data durability. For more information about NVMe commands, see NVM Express Base
 Specification.
- Cross-instance data sharing: This feature enables data sharing across multiple ECS instances to reduce storage costs and improve service flexibility.
- Disaster recovery: This feature allows quick scheduling of services to normal ECS instances to ensure service continuity in single-point-of-failure (SPOF) scenarios.

Limits

The following limits apply to the multi-attach feature:

- The following instance families support the feature.
- ecs.ebmg7s-se-x25-c1m8
- ecs.ebmg7m-se-x25-c1m8
- ecs.ebmg7x-se-x25-c1m8
- ecs.ebmg7s-se-numaoff-x25-c1m8
- ecs.ebmg7m-se-numaoff-x25-c1m8
- ecs.ebmg7x-se-numaoff-x25-c1m8
- ecs.g7x-se-x25
- Data disks are supported but system disks are not supported.
- The multi-attach feature can be enabled only when you create premium performance disks. This feature cannot be enabled or disabled after premium performance disks are created.
- After the multi-attach feature is enabled for a premium performance disk, we recommend that you use cluster file systems such as Oracle Cluster File System version 2 (OCFS2), Global File System 2 (GFS2), Veritas Cluster File System (Veritas CFS), Oracle Automatic Storage Management Cluster File System (Oracle ACFS), and Databricks File System (DBFS) on the disk.

🔥 Warning

When a premium performance disk for which the multi-attach feature is enabled is attached to multiple ECS instances, data cannot be synchronized among the instances and data inconsistency may occur if file systems such as EXT2, EXT3, EXT4, XFS, and New Technology File System (NTFS) are used.

• The performance of premium performance disks is limited. When a premium performance disk is attached to multiple ECS instances, the total performance of the disk on all instances cannot exceed the maximum performance that can be provided by the disk.

Premium performance disks for which the multi-attach feature is enabled are subject to the functionality limits described in the following table.

Feature	Limits
Disk attaching	A single premium performance disk can be attached to up to 16 ECS instances that support the NVMe protocol.
Release of disks with ECS instances	Not supported.
Disk re-initialization	Not supported.

Disk category change	Not supported.
Disk resizing	Only offline resizing of disks is supported. For more information, seeResize a disk.
Snapshot-consistent group	Not supported.

Step 1: Create a disk for which the multi-attach feature is enabled

To attach a disk to multiple ECS instances, you must enable the multi-attach feature for the disk when you create the disk.

1. Log on to the ECS console.

i. Log on to the Apsara Uni-manager Management Console.

ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.

2. In the left-side navigation pane, choose Storage & Snapshots > Disks.

3. In the top navigation bar, select an organization, a resource set, and a region.

4. Click Create Disk.

5. On the Create Disk page, set the parameters for creating a disk. The following table describes the parameters.

Section	Parameter	Required	Description		
	Organization	Yes	The organization to which the disk belongs.		
Aroa	Resource Set	Yes	The resource set to which the disk belongs.		
Alea	Region	Yes	The region in which the disk resides.		
	Zone	Yes	The zone in which the disk resides.		
	Creation Method	Yes	The method to create the disk. Valid values: Disk Creation and Storage Set Creation . If you want to create the disk in a partition of a specified storage set, select Storage Set Creation .		
			 If no storage set is available in the current environment, you can select only Disk Creation. You must configure a storage set before you can select Storage Set Creation. Before you create a disk in a partition of a storage set, make sure that the storage set is created and the partition is configured. For more information about how to create a storage set, see Create a storage set. 		
	Storage Set	Yes	The name of the storage set. This parameter is required only if you set the Creation Method parameter to Storage Set Creation.		
	Partitions	Yes	The number of partitions in the storage set. The number must be greater than or equal to 2. This parameter is required only if you set the Creation Method parameter to Storage Set Creation .		
Basic Configurations	Name	Yes	The name of the disk. The name must be 2 to 128 characters in length. It must start with a letter and cannot start with http:// or https://. It can contain letters, digits, periods (.), underscores (_), hyphens (-), colons (:), and commas (,). If you create multiple disks at a time, the system automatically adds numbers such as 001 and 002 to the specified disk name. For example, if you set the Name parameter to mydisk and create two disks at a time, the names of the created disks are mydisk001 and mydisk002.		
	Quantity	Yes	The number of disks to create. You can create multiple disks at a time. Valid values: 1 to 100. Default value: 1.		
	Specifications	Yes	The category of the disk. In this topic, Ultra Disk is selected. The disk size ranges from 20 GiB to 32,768 GiB.		
	Multi-attach	No	Specifies whether to enable the multi-attach feature for the disk. In this topic, $\ensuremath{\text{Yes}}$ is selected.		
	Encryption	No	Specifies whether to encrypt the disk. Default value: No.		
	Use Snapshot	No	 Specifies whether to create the disk based on a snapshot. If you selectYes, you must specify a snapshot. The size of the created disk depends on the size of the specified snapshot. If the disk size that you specify is greater than the snapshot size, the disk is created with the size that you specify. If the disk size that you specify is smaller than the snapshot size, the disk is created with the snapshot size. 		

⑦ Note For more information, see Create a disk.

6. Confirm the configurations and click Submit.

In the message that appears, click OK. After the disk is created, return to the Disks page. You can view the new disk. The status of the disk is Available, and Enabled is displayed in the Multi-attach column that corresponds to the disk.

Step 2: Attach the disk to multiple ECS instances that support the NVMe protocol

Before you attach the disk to ECS instances, make sure that the following requirements are met:

- The disk and the ECS instances reside in the same zone.
- The instance families and images of the ECS instances comply with the NVMe protocol. For more information, see Limits.
- Obtain the ID of an ECS instance to which you want to attach the disk. In the left-side navigation pane, choose Instances & Images > Instances. On the Instances page, view and copy the instance ID in the instance list.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Disks.
- 3. Find the disk, click the icon in the Actions column, and then select Attach.
- 4. In the Attach dialog box, select the ID of the ECS instance obtained in Step 1 from the Instance drop-down list and click OK.

⑦ Note You can attach the disk to only a single instance each time you perform the preceding steps. To attach the disk to multiple ECS instances, repeat the preceding steps.

After the disk is attached, you can check that the status of the disk changes to **Running** on the **Disks** page. You can also view the attachment information of the disk on one or more ECS instances in the **Attachment Information** section on the details panel of the disk.

3.4.6. Encrypt a disk

3.4.6.1. Encrypt a system disk

In the scenarios that require data security and regulatory compliance, you can encrypt disks to secure your data stored in ECS. To encrypt system disks, you can encrypt custom images and then use the encrypted custom images to create instances. The system disks of the created instances are automatically encrypted.

Background information

You can encrypt system disks only by encrypting custom images.

Step 1: Create a custom image

1. Log on to the ECS console.

- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose **Storage & Snapshots > Snapshots**.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. On the Disk Snapshots tab, find the snapshot that you want to manage and click Create Custom Image in the Actions column.
- 5. In the Create Custom Image dialog box, set the parameters as required. The following table describes the parameters.

Parameter	Description
System Snapshot ID	The ID of the snapshot. The system automatically obtains the snapshot ID and you cannot modify it.
Sharing Scope	The scope in which to share the custom image. Valid values: Current Organization and Subordinate Organizations Current Resource Set Current Organization
Image Name	The name of the custom image. The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), hyphens (-), and colons (:). The name cannot start with a special character or a digit.
Image Description	The description of the custom image. The description must be 2 to 256 characters in length and cannot start with http:// or https://.

6. Click **OK**.

Step 2: Encrypt the custom image

- 1. Log on to the ECS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.
- 2. In the left-side navigation pane, choose **Instances & Images > Images**.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. On the Custom Images tab, find the custom image that you created, move the pointer over the ... icon in the Actions column, and then select

Encrypt Image

5. In the Encrypt Image dialog box, set the parameters as required. The following table describes the parameters.

Parameter	Description
Image ID	The ID of the custom image. The system automatically obtains the ID of the image and you cannot modify it.
Name	The name of the encrypted custom image. The name must be 2 to 128 characters in length. It must start with a letter and can contain only the following special characters: underscores (_), periods (.), and hyphens (-).
Description	The description of the encrypted custom image. The description must be 2 to 256 characters in length and cannot start with http:// or https://.

6. Click **OK**.

Step 3: Use the encrypted custom image to create an instance

- 1. Log on to the ECS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose **Instances & Images > Instances**.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Click Create Instance.
- 5. Configure the properties of the instance.
 - For more information about how to configure instance properties, see the Create an instance section in ECS User Guide.

In the **Image** section, set the Image Type parameter to **Custom Image**. Then, select the encrypted custom image from the **Custom Image** dropdown list.

6. Click Submit.

Result

After the instance is created, you can click its ID to go to the **Instance Details** page. Then, you can click the **Disks** tab and check the value in the **Encrypted** column corresponding to the system disk. If the value is **Yes**, the system disk is encrypted.

3.4.6.2. Encrypt a data disk

In the scenarios that require data security and regulatory compliance, you can encrypt disks to secure your data stored in ECS. After a data disk is created, you cannot change its encryption state. If you want to encrypt a data disk, enable encryption for the disk when you create it.

Procedure

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Disks.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Click Create Disk.
- Configure the disk based on the following description.
- When you create the disk, take note of the following parameter settings:
- Encryption: Select Yes.
- Encryption Method: Select an encryption algorithm. Valid values:
- AES256.
- Encryption Key: Select an encryption key.

For more information about disk configuration parameters, seeCreate a disk.

6. Click Submit.

3.4.7. Roll back a disk by using a snapshot

If you have created a snapshot for a disk, you can use the rollback feature to restore the data of the disk to a specific point in time. The disk rollback operation is irreversible. After the disk is rolled back, the data stored on the disk before the operation is performed cannot be recovered. Exercise caution when you perform this operation.

Prerequisites

- A snapshot of the disk to be rolled back is created, and no other snapshot is being created for the disk. For more information, see Create a snapshot.
- The disk is not released.
- The instance to which the disk belongs is in the Stopped state.

Procedure

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Snapshots.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find the snapshot that you want to use to roll back the specified disk and click Roll Back Disk in the Actions column.
- 5. In the message that appears, click Roll Back Disk.

What to do next

- After a disk is rolled back, the host configuration file and the configurations, such as the hostname, SSH key pair, password, network, system source, and clock source, are initialized. You must reconfigure the settings.
- If you resized a disk after you created a snapshot for the disk, the size of the disk is also rolled back after you roll back the disk. To make the disk revert to the size before the rollback, you must log on to the instance to resize the file system. For more information, see Resize a disk.

3.4.8. Re-initialize disks

3.4.8.1. Re-initialize a system disk

This topic describes how to re-initialize a system disk. After a system disk is re-initialized, the system disk is restored to the state at the time when it was created.

Prerequisites

- The disk is in the **Running** state.
- The instance is in the **Stopped** state.
- After a disk is re-initialized, the data stored on the disk is deleted and cannot be recovered. Exercise caution when you perform this operation. If you

need the data stored on the system disk, we recommend that you back up the disk data or create snapshots before you re-initialize the disk. For more information, see Create a snapshot.

Background information

The result of disk re-initialization depends on the disk type and how the disk was created.

If a system disk is attached to an ECS instance, you can re-initialize the system disk to restore it to the state at the time when it was created. The following section describes the changes of a system disk after the disk is re-initialized:

• The system disk is restored to the state at the time when it was created.

▲ Warning After a disk is re-initialized, all data stored on the disk is deleted. We recommend that you create snapshots for a disk to back up data before you re-initialize the disk.

- The automatic snapshot policy applied to the system disk before the re-initialization remains valid.
- The IP addresses and disk IDs of the instance remain unchanged.
- The automatic and manual snapshots of the system disk remain available. You can use these snapshots to roll back the disk. For more information, see Roll back a disk by using a snapshot.

Procedure

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose **Storage & Snapshots > Disks**.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find the system disk that you want to re-initialize and click Re-initialize Disk in the Actions column.
- 5. In the **Re-initialize Disk** dialog box, set the parameters as required. The following table describes the parameters.

Parameter	Description	
Password	Reset the password that is used to log on to the instance. You can use the old password or specify a new one.	
Confirm Password		
Instance Startup Policy	If you select Start Instance After Re-initializing Disk, the instance automatically starts after you re-initialize the system disk.	

6. Click **OK**.

Result

When the disk is being re-initialized, the disk is in the Initializing state. After the disk is re-initialized, it enters the Running state.

What to do next

If data disks have been attached to a Linux instance before you re-initialize its system disk, you must re-create the mount targets for the data disk
partitions and mount file systems to the data disk partitions.

O Note After you re-initialize the system disk of a Linux instance, data stored on the data disks attached to the instance remains unchanged, but the attachment information of the data disks is lost. You must re-create the mount targets for the data disk partitions and mount file systems to the data disk partitions.

- After the system disk is re-initialized, you must redeploy applications and reconfigure the settings on the disk to restore your business as soon as possible.
- If you have created snapshots for the system disk before the system disk is re-initialized, you can use the snapshots to create data disks and attach the data disks to the ECS instance to obtain data originally stored on the system disk. For more information, see Create a snapshot.

3.4.8.2. Re-initialize a data disk

If a data disk is attached to an ECS instance, you can re-initialize the disk to restore it to the state at the time when it was created.

Prerequisites

- Snapshots are created for the data disk. For more information, see Create a snapshot.
- The data disk is attached to an ECS instance. For more information, see Attach disks.
- The ECS instance is in the **Stopped** state.

For a Linux instance, if you create an empty data disk and add a command in the /etc/fstab file to mount partitions of the data disk at the system
startup, the command is not executed and the instance cannot start as expected after you re-initialize the data disk. In this case, you must comment
out the command in the /etc/fstab file. To do so, perform the following operations:

- i. Connect to the instance. For more information, see the Connect to instances section in ECS User Guide.
- ii. Run the vim /etc/fstab command.
- iii. Press the I key to enter the edit mode.
- iv. Find the command used to mount data disk partitions and comment it out by adding a number sign (i) to the beginning of the command line.

/dev/vdb1 /InitTest ext3 defaults 0 0

(2) Note /dev/vdb1 is a disk data partition and /InitTest is a mount target used in this example. You can modify the command based on your business requirement.

v. Press the ESC key to exit the edit mode. Then, enter ing to save the file and exit.

Background information

The state of a data disk after it is re-initialized varies based on its initial state when it was created and the operating system that the instance runs:

- The data disk is restored to the initial state when it was created:
 - $\circ~$ The data disk becomes an empty disk if it was originally an empty disk.
 - The data disk stores the data recorded in the source snapshot if the data disk was created from a snapshot.

User Guide•EBS

- For a Windows instance, after you re-initialize a data disk, the data disk is ready for use without the need for additional operations, regardless of its initial state.
- For a Linux instance:
- If a data disk was created from a snapshot, the data disk stores only the data recorded in the source snapshot after the data disk is re-initialized. You do not need to mount the partitions again, but all the data generated after the disk was created is lost.
- If a data disk was created as an empty disk, all the data and file systems on the disk are lost after the data disk is re-initialized. You must reformat
 and partition the disk, and then mount the partitions again.

Procedure

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Disks.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find the data disk that you want to re-initialize and click Re-initialize Disk in the Actions column.
- 5. In the **Re-initialize Disk** dialog box, click **OK**.

What to do next

- For a Linux instance, if the data disk was created as an empty disk, you must format the data disk after you re-initialize it. For more information, see Format a data disk on a Linux instance.
- After the data disk is re-initialized, you must redeploy applications and reconfigure the settings on the disk to restore your business as soon as possible.

3.4.9. Change the category of a disk

Alibaba Cloud offers different categories of disks to meet the storage performance and cost requirements for a variety of scenarios. You can change the categories of your disks based on your needs. For example, assume that you have created a standard SSD. If you want the standard SSD to deliver a higher IOPS, you can change this standard SSD into an enhanced SSD (ESSD).

Prerequisites

Snapshots of the disk whose category you want to change are created. For more information, see Create a snapshot for a disk.

Background information

For information about the performance of different disk categories, see Block storage performance.

Before you change the category of a disk, take note of the following items:

• Only the disk category change operations described in the following table are supported.

Category of the original disk		Description of supported category change	
Ultra disk		You can convert ultra disks into standard SSDs, PL0 ESSDs, PL1 ESSDs, PL2 ESSDs, or PL3 ESSDs. ③ Note You cannot convert ultra disks into standard SSDs in Hangzhou Zone D.	
Standard SSD		You can convert standard SSDs into PL1, PL2, or PL3 ESSDs.	
	PL0	You can convert PL0 ESSDs into PL1, PL2, or PL3 ESSDs.	
ESSD	PL1, PL2, and PL3	 You can convert PL1, PL2, and PL3 ESSDs based on their billing methods. If ESSDs use the pay-as-you-go billing method, you can convert the ESSDs between PL1, PL2, and PL3. If ESSDs use the subscription billing method, you can upgrade the ESSDs only from the low performance level to the high performance level. You can upgrade PL1 ESSDs to PL2 ESSDs. You can upgrade PL1 ESSDs to PL3 ESSDs. You can upgrade PL2 ESSDs to PL3 ESSDs. 	

? Note

• Before you convert a disk on an Elastic Compute Service (ECS) instance into an ESSD, check whether the instance type supports ESSDs. For more information, see Overview of instance families.

If the instance type does not support ESSDs, you must upgrade the instance to an instance type that supports ESSDs and then convert the disk. For more information, see Instance families that support instance type changes.

The performance levels to which ESSDs can be upgraded are determined based on their capacity. If you cannot select a higher
performance level for an ESSD, resize the ESSD and then upgrade its performance level. For more information, see Overview.

• When you change the category of a disk, the performance of the disk may also change. We recommend that you change the categories of your disks during off-peak hours.

• It may take an extended period of time to change the category of a disk. The amount of time required is determined based on the storage capacity, original category, and throughput of the disk. After you submit the request to change the category of a disk, we recommend that you view the change progress by going to the **Task logs** page in the ECS console or by calling the DescribeTaskAttribute operation.

• Disk categories may fail to change if resources are insufficient. If a failure occurs, try again later.

Limits

Phase	Limits
Before the category change	Before you change the category of a disk, make sure that the following requirements are met:Up to five disk category change tasks can be concurrently performed within the same region and account.You cannot change the category of a disk attached to an instance that is included in a migration plan.

During the category change	 The following limits apply while the category of a disk is being changed: You cannot cancel the disk category change task after it is initiated. You cannot create snapshots for the disk. You cannot resize the disk. You cannot partition or format the disk. You cannot re-initialize the disk. You cannot roll back the disk by using snapshots. You cannot attach the disk to an instance or detach it from an instance. If the disk is an ESSD, you cannot change its performance level. If the disk is a system disk, you cannot replace the operating system of the instance.
After the category change	The following limits apply after the category of a disk is changed:If only the performance level of an ESSD is changed, you can change the ESSD category again at any time.A disk can have its category changed only once within seven days.

Billing

After the category of a disk is changed, the billing of the disk has the following changes:

- After the category of a pay-as-you-go disk is changed, you are charged for the disk based on the new disk category.
- After the category of a subscription disk is changed, the price difference is calculated based on the remaining subscription duration and the new disk category. You must pay for the difference.

For information about disk pricing, see Block storage devices.

Procedure

- 1. In the left-side navigation pane, choose Storage & Snapshots > Block Storage (Disks) .
- 2. In the top navigation bar, select the region and resource group to which the resource belongs.

C→ Alibaba Cloud 🗠 Workbench 🗏 ProdResourceGroup 🗸 💇 China (Beijing) 🗸

- 3. Find the disk whose category you want to change and choose More > Change Category in the Actions column.
- 4. In the Change Disk Category dialog box, set New Disk Category.

Once After you convert a disk to an PL3 ESSD, you must attach the disk to an instance or restart the instance to which the disk is attached so that the PL3 ESSD can deliver optimal performance. Otherwise, the PL3 ESSD cannot deliver optimal performance but data reliability is not affected.

5. Confirm the price and click OK.

Results

- Use one of the following methods to view the change progress and the state of the disk:
- Method 1: Go back to the Disks page and view the category of the disk in the Disk Category column.
- Method 2: In the left-side navigation pane, choose Maintenance & Monitoring > Tasks. Find the change task and view the change progress in the Status column.

Related topics

- ModifyDiskSpec
- DescribeDisks
- DescribeTaskAttribute

3.4.10. Detach a data disk

You can detach a data disk from an ECS instance if the disk is no longer needed. You must detach a data disk from an instance before you can attach the disk to another instance in the same zone. This topic describes how to detach a data disk.

Prerequisites

Before you detach a data disk, make sure that the following conditions are met:

- The disk is attached to an instance and in the Running state.
- To avoid data loss and ensure data integrity, we recommend that you suspend read and write operations on the disk before you detach it.

Background information

To detach a data disk, follow the following procedure.

(?) Note Local disks that are used as data disks cannot be detached.

1. If file systems are mounted to the partitions of the data disk, detach the data disk on the operating system of the instance to which the data disk is attached.

For more information, see Step 1: Detach the data disk on the operating system of the instance to which the disk is attached.

2. On the Disks page, detach the data disk.

For more information, see Step 2: Detach the data disk on the Disks page.

Step 1: Detach the data disk on the operating system of the instance to which the disk is attached

If the data disk is partitioned and has file systems mounted, follow one of the following procedures to detach the data disk on the operating system of the instance to which the disk is attached.

Procedure for a Linux instance

- 1. Connect to the ECS instance. For more information, see the Connect to instances section in **ECS User Guide**.
- 2. Run the following command to view the mount information of the data disk:

df -h

A command output similar to the following one is displayed. In this example, the /dev/vdb1 partition of the data disk is used. In actual scenarios, choose the data disk partitions whose mount information you want to view.

lroot@ecs]#	di —h					
Filesystem		Size	Used	Avail	Use%	Mounted	on
devtmpfs		441M		441M	0%	/dev	
tmpfs		459M		459M	0%	/dev/shm	ι I
tmpfs		459M	468K	459M	1%	/run	
tmpfs		459M	0	459M	0%	/svs/fs/	'cgrou
Iday Indat		400	2.60	350	76	1	
dev/vdb1		40G	49M	38G	1%	/mnt	
tmnts		920	U	920	USh	/run/use	er /0

 Run the umount command to unmount the file systems from the data disk partitions. For example, you can run the following command to unmount the file system from the /dev/vdb1 partition of the sample data disk:

umount /dev/vdb1

4. Run the following command to view the universally unique identifier (UUID) information of the data disk partitions:

blkid

The following command output shows the UUID information of the /dev/vdb1 partition of the data disk.

[root@ecs ~]# blkid		
/dev/vda1: UUID="9f2d3e15-	-0165b4b67864"	TYPE="ext4"
/dev/vdb1: UUID="430d44fe-	-5d01bc597839"	TYPE="ext4"
[root@ecs ~]#		

5. Run the following command to check whether the /etc/fstab file contains the automatic mount configuration information of the file systems in the data disk partitions:

cat /etc/fstab

Find the UUID information obtained in the previous step in the command output. The file system mounted to the /dev/vdb1 partition is configured in /etc/fstab, as shown in the following figure.



6. Delete the automatic mount configuration information of the file systems in the data disk partitions from /etc/fstab.

③ Note If you do not delete the automatic mount configuration information of the file systems in the data disk partitions from/etc/fstab, the instance cannot be restarted after you detach the data disk from the instance in the ECS console.

i. Run the following command to open /etc/fstab:

vim /etc/fstab

- ii. Press the I key to enter the edit mode.
- iii. Delete or comment out the automatic mount configuration information of the file systems in the data disk partitions. For example, you can add a number sign (±) to the beginning of the line of the automatic mount configuration information to comment the information out.



iv. Press the ESC key, enter in and then press the ENTER key to save your edits and exit the edit mode.

Procedure for a Windows instance

⑦ Note In this example, Windows Server 2012 R2 is used.

- 1. Connect to the ECS instance.
- For more information, see the Connect to instances section in ECS User Guide.
- 2. On the Windows Server desktop, click the Server Manager icon in the lower-left corner.



- 3. In the upper-right corner of the Server Manager window, choose Tools > Computer Management.
- 4. In the left-side navigation pane, choose Computer Management (Local) > Storage > Disk Management
- 5. Right-click the disk that you want to detach and select Offline.

Disk 2	
39.88 GB Online	New Spanned Volume New Striped Volume New Mirrored Volume New RAID-5 Volume
	Convert to Dynamic Disk Convert to MBR Disk
	Offline
	Properties
	Help

Step 2: Detach the data disk on the Disks page

You can detach the data disk on the Instances or Disks page. To detach the data disk on the Disks page in the ECS console, perform the following steps: 1. Log on to the ECS console.

- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose **Storage & Snapshots > Disks**.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find the disk that you want to detach, move the pointer over the ____ icon in the Actions column, and then select Detach.

5. In the message that appears, click **Detach**.

You can also detach a data disk on the **Disks** tab of the details page of an instance.

Result

After you perform the preceding steps to detach the data disk, you can choose **Storage & Snapshots > Disks** in the left-side navigation pane of the ECS console and then find the disk on the Disks page. If **Available** is displayed in the **Status** column, the disk is detached from the original instance.

What to do next

- You can attach the data disk to another instance in the same zone. For more information, see Attach disks.
- If the data disk is no longer needed, you can back up the data stored on the disk and then release the disk. For more information, see Create a snapshot and Release a data disk.

3.4.11. Release a data disk

You can manually release the data disks that you no longer need. If a data disk is released, all data stored on the disk is also released. This topic describes how to release a data disk on the Disks page in the ECS console.

Prerequisites

The data disk to be released is in the **Available** state. If the data disk is attached to an instance, you must detach the disk from the instance before you can release the disk.

Procedure

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Disks
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find the disk that you want to release, move the pointer over the _____ icon in the Actions column, and then select Release.

5. In the message that appears, click **Release**.

3.5. Snapshots

3.5.1. Disk snapshots

3.5.1.1. Create a snapshot

You can use snapshots to back up data, restore ECS instances that were accidentally released, and create custom images. You can create snapshots of disks to improve fault tolerance before you roll back a disk, modify key system files, or change the operating system of an instance.

Prerequisites

- The associated instance of the disk for which you want to create a snapshot is in the Running or Stopped state.
- The disk is in the **Running** state.

Background information

Up to 64 snapshots can be retained for each disk.

Snapshots can be used in the following scenarios:

- Roll back the data of the disk for which a snapshot is created. For more information, see Roll back a disk by using a snapshot.
- Create a custom image. For more information, see Create a custom image from a snapshot.
 - () **Important** You cannot use a data disk snapshot to create custom images.
- Create a data disk from a data disk snapshot.

To create a data disk from a data disk snapshot, set the Use Snapshot parameter to Yes and then specify a snapshot on the Create Disk page. For more information, see Create a disk. The size of the created disk is determined by the size of the specified snapshot and cannot be changed. When you reset such a disk, the disk data is also restored to the data of the snapshot at the time when the disk was created.

When you create a snapshot, take note of the following items:

• For each disk, the first snapshot is a full snapshot and subsequent snapshots are incremental snapshots. It takes longer to create the first snapshot. The amount of time it takes to create an incremental snapshot depends on the volume of data that has been changed since the last snapshot. The more data that has changed, the longer it takes.

② Note If you want to use a snapshot immediately after it is created, you can enable the instant access feature

· We recommend that you create snapshots during off-peak hours.

Procedure

1. Log on to the ECS console.

- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Disks.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find the disk for which you want to create a snapshot and click Create Snapshot in the Actions column.
- 5. In the Create Snapshot dialog box, set the parameters and click OK.

Parameter	Description		
	The name of the snapshot. The name must be 2 to 128 characters in length. It must start with a letter and cannot start with http:// or https://. The name can contain digits, letters, colons (:), underscores (_), and hyphens (-).		
Snapshot Name	⑦ Note The name cannot start with auto because snapshots whose names start with auto are recognized as automatic snapshots.		
Construction Description	The description of the snapshot. The description must be 2 to 256 characters in length and cannot start with		
Snapsnot Description	http:// or https://.		

You can go to the Snapshots page to check the creation progress of the snapshot. For more information, see Search for and view a snapshot. After the snapshot is created, 100% is displayed in the Progress column.

3.5.1.2. Search for and view a snapshot

You can search for and view the created snapshots and their details.

Procedure

- 1. Log on to the ECS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose **Storage & Snapshots > Snapshots**.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. On the **Disk Snapshots** tab, use one of the following methods to search for the required snapshot:
 - Select a field from the field drop-down list and enter a keyword in the search box. Then, click the Q icon. The snapshots that meet the filter condition are displayed in the snapshot list.
 - Click Advanced Filter. Specify values for different fields as required and click Search. The snapshots that meet the filter conditions are displayed in the snapshot list.

③ Note When you use the advanced filtering feature, you can specify multiple filter conditions to narrow down search results.

Field	Description
Snapshot Name	Enter a complete snapshot name or a keyword.
Snapshot ID	Enter a complete snapshot ID or a keyword.
Disk ID	Enter the ID of a disk to search for the snapshots related to the disk.
Creation Time	Specify a point in time or a time range to search for the snapshots that meet the condition.
Tag	Select a tag key or a tag value to search for the snapshots to which the tag is added.

3.5.1.3. Roll back a disk by using a snapshot

If you have created a snapshot for a disk, you can use the rollback feature to restore the data of the disk to a specific point in time. The disk rollback operation is irreversible. After the disk is rolled back, the data stored on the disk before the operation is performed cannot be recovered. Exercise caution when you perform this operation.

Prerequisites

- A snapshot of the disk to be rolled back is created, and no other snapshot is being created for the disk. For more information, see Create a snapshot.
- The disk is not released.
- The instance to which the disk belongs is in the Stopped state.

Procedure

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Snapshots.
- 3. In the top navigation bar, select an organization, a resource set, and a region.

4. Find the snapshot that you want to use to roll back the specified disk and click Roll Back Disk in the Actions column.

5. In the message that appears, click **Roll Back Disk**.

What to do next

- After a disk is rolled back, the host configuration file and the configurations, such as the hostname, SSH key pair, password, network, system source, and clock source, are initialized. You must reconfigure the settings.
- If you resized a disk after you created a snapshot for the disk, the size of the disk is also rolled back after you roll back the disk. To make the disk revert to the size before the rollback, you must log on to the instance to resize the file system. For more information, see Resize a disk.

3.5.1.4. Create a custom image from a snapshot

You can create a custom image from a snapshot that contains the operating system and data of an ECS instance. Then, you can use the custom image to create multiple identical instances.

Prerequisites

A system disk snapshot is created. For more information, see Create a snapshot.

Background information

Before you create custom images from snapshots, take note of the following items:

- Notes on snapshots used to create custom images:
 - A custom image can be created from a system disk snapshot or from a system disk snapshot and one or more data disk snapshots. Data disk snapshots alone cannot be used to create custom images.
 - Both encrypted and unencrypted snapshots can be used to create custom images.
- Notes on custom images: Custom images cannot be used across regions.

Procedure

- 1. Log on to the ECS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Snapshots.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find the snapshot of a disk whose value in the Disk Type (All) column is System Disk and click Create Custom Image in the Actions column.
- 5. In the Create Custom Image dialog box, set the parameters as required. The following table describes the parameters.

Parameter	Description
Sharing Scope	The scope in which to share the custom image. Valid values: Current Organization and Subordinate Organizations Current Resource Set Current Organization
Image Name	The name of the custom image. The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), hyphens (-), and colons (:). The name cannot start with a special character or a digit.
Image Description	The description of the custom image. The description must be 2 to 256 characters in length and cannot start with http:// or https://.

6. Click **OK**.

3.5.1.5. Delete one or more snapshots at a time

In the ECS console, you can delete a snapshot that is no longer needed.

Prerequisites

- Deleted snapshots cannot be restored. Proceed with caution.
- If a system disk snapshot has been used to create a custom image, the snapshot cannot be deleted.

Procedure

- 1. Log on to the ECS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.
- 2. In the left-side navigation pane, choose **Storage & Snapshots > Snapshots**.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Use one of the following methods to delete one or more snapshots as required:
- To delete a single snapshot, find the snapshot and click **Delete** in the **Actions** column.
- To delete multiple snapshots at a time, select the snapshots and click Delete in the lower-left corner of the Snapshots page.
- 5. In the message that appears, click Delete

3.5.2. Snapshot-consistent groups

3.5.2.1. Create a snapshot-consistent group

You can create a snapshot-consistent group to simultaneously create snapshots for disks on an ECS instance. If a business system involves multiple disks, you can create a snapshot-consistent group to ensure time series consistency of data writes and crash consistency of the data in the business system.

Background information

You can use snapshot-consistent groups to simultaneously create snapshots for multiple disks on an ECS instance. Snapshot-consistent groups can be applied to cluster business. The following section describes the typical scenarios:

- A business system is deployed on multiple disks of an ECS instance, and time series consistency and crash consistency are required across databases or enterprise-level applications. For example, a MySQL cluster is built on multiple ECS instances, a single Logical Volume Manager (LVM) logical volume is created across multiple volumes, or Oracle or SAP HANA clusters are migrated to the cloud.
- Snapshots need to be created simultaneously for multiple nodes of a distributed application system, such as a large-scale website or a multiapplication collaborative system.
- Disks on an ECS instance need to be batch backed up and high time series consistency is required.

Precautions

Before you use the snapshot-consistent group feature, take note of the following items:

- Notes on the creation of a snapshot-consistent group based on an instance:
 To ensure that services are not affected, we recommend that you create the snapshot-consistent group during off-peak hours.
- The instance must be in the **Running** or **Stopped** state.
- Only standard performance disks and premium performance disks are supported. The disks involved must be in the In Use state. If the instance
 has disks of other categories attached, you can select only standard performance disks or premium performance disks on the instance to create a
 snapshot-consistent group.
- A single snapshot-consistent group can contain snapshots of up to 16 disks, including the system disks and data disks, and the total size of the disks cannot exceed 32 TiB.
- Snapshots that you created are retained until they are deleted. We recommend that you delete unnecessary snapshots on a regular basis to control the total size. For more information, see <u>Delete one or more snapshots at a time</u>.
- Notes on the amount of time required to create a snapshot-consistent group:

The amount of time required to create a snapshot-consistent group depends on the total size of the selected disks for which to create snapshots. The first snapshot of each disk is a full snapshot. It takes longer to create the first snapshot than the subsequent snapshots. Subsequent snapshots are incremental snapshots. The amount of time required varies based on the amount of data changed since the previous snapshot. If you want to use a snapshot immediately after it is created, you can enable the instant access feature.

Procedure

- 1. Log on to the ECS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Snapshots.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. On the Snapshot-consistent Groups tab, click Create Snapshot-consistent Group.
- 5. In the Create Snapshot-consistent Group dialog box, set the parameters as required and click OK. The following table describes the parameters.

Parameter	Description
Data Source	Select the instances based on which to create the snapshot-consistent group.
Disks	Select standard performance disks and premium performance disks for which you want to create the snapshot- consistency group. Up to 16 disks can be selected, including the system disk and data disks. The total size of the selected disks cannot exceed 32 TiB.
Group Name	 Enter a name for the snapshot-consistent group for easy management. The name must meet the following requirements: The name must be 2 to 128 characters in length. It must start with a letter and cannot start with http:// or <a href="http://. It can contain letters, digits, periods (.), underscores (_), hyphens (-), and colons (:).
Description	Enter a description for the snapshot-consistent group for easy management. The description must be 2 to 256 characters in length and cannot start with http:// or https://.
Instant Access	 The instant access feature can accelerate the process of snapshot creation. You can use the instant access feature to create snapshots within seconds. Enable Instant Access: You can select this check box to enable the instant access feature. By default, this feature is disabled. Snapshot Speed Available Duration: After you select the checkbox, you can specify a validity duration for the feature. The default validity duration is one day. The instant access feature is automatically disabled when the specified duration expires.

What to do next

After a snapshot-consistent group is created, you can use the snapshot-consistent group to roll back disks based on your business requirements. For more information, see Roll back disks by using a snapshot-consistent group.

3.5.2.2. Search for and view a snapshot-consistent group

You can search for the created snapshot-consistent groups and view their details.

Procedure

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose **Storage & Snapshots > Snapshots**.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. On the **Snapshot-consistent Groups** tab, select a field from the field drop-down list and enter a keyword in the search box. The system then displays the snapshot-consistent groups that meet the condition.

Field

Description

Snapshot-consistent Group Name	Enter the name of a snapshot-consistent group to search for the snapshot-consistent group.
Snapshot-consistent Group ID	Enter the ID of a snapshot-consistent group to search for the snapshot-consistent group.

What to do next

After you find the required snapshot-consistent group, you can view the ID, name, status, and type of the snapshot-consistent group, whether the instant access feature is enabled for the group, and the instance to which the group belongs.

You can click the ID or name of the snapshot-consistent group to go to its details page. On this page, you can view the basic information of the snapshot-consistent group. You can also view the ID, name, status, disk properties, disk capacity, and encryption setting of each snapshot in the group.

3.5.2.3. Roll back disks by using a snapshot-consistent group

After a snapshot-consistent group is created, you can use it to roll back one or more disks in the event of system failures or data errors caused by accidental operations.

Prerequisites

Before you use a snapshot-consistent group to roll back disks on instances, make sure that the following requirements are met:

• A snapshot-consistent group is created based on the instances. For more information, see Create a snapshot-consistent group.

() Important The rollback operation cannot be reversed. When a disk is rolled back, if data changes were made to the disk from the time the snapshot-consistent group was created until the disk is rolled back, all these data changes are lost. To prevent data loss caused by accidental operations, we recommend that you create a snapshot-consistent group to back up the disk data before you roll back disks.

- The instances are in the Stopped state. For more information, see the Stop an instance section under Instance > Manage instance status in ECS User Guide.
- The disks that correspond to the snapshots contained in the snapshot-consistent group have not been released or detached or do not have snapshots being created.
- The operating systems of the instances have not been replaced since the snapshot-consistent group was created.
- The snapshots of the disks have not been deleted from the snapshot-consistent group. If the snapshot of a disk has been deleted from the snapshot-consistent group, the disk cannot be rolled back by using the snapshot-consistent group. The snapshot-consistent group can be used to roll back only other disks whose snapshots are contained in the group.

Procedure

- 1. Log on to the ECS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Snapshots.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. On the Snapshot-consistent Groups tab, find the snapshot-consistent group that you want to use and click Roll Back in the Actions column.
- 5. In the **Roll Back** dialog box, perform a rollback operation.
 - i. Read the notes on rolling back disks on instances and determine whether to select Start Instance Immediately After Rollback based on your business requirements.
- ii. In the **Disk Snapshots** section, select the snapshots of the disks that you want to roll back.
- iii. Click Roll Back.
- After the selected disks are rolled back, the Rollback successful message is displayed.

Result

After disks are rolled back, you can log on to their associated instances to check whether the disk data has been reverted to the state at the time when the snapshots were created.

3.5.2.4. Delete a snapshot-consistent group

You can delete a snapshot-consistent group if it is no longer used.

Background information

When you delete a snapshot-consistent group, the system does not delete the disk snapshots that have been used to create custom images. You must delete the images that are created by using the disk snapshots before you can delete the disk snapshots.

Procedure

- 1. Log on to the ECS console.
 - i. Log on to the Apsara Uni-manager Management Console.
 - ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Snapshots.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. On the Snapshot-consistent Groups tab, use one of the following methods to delete one or more snapshot-consistent groups as required:
 - To delete a single snapshot-consistent group, find the snapshot-consistent group and click Delete in the Actions column.
 - To delete multiple snapshot-consistent groups at a time, select the snapshot-consistent groups and click **Delete** in the lower part of the snapshot-consistent group list.
- 5. In the message that appears, click **Delete**.

3.6. Automatic snapshot policies

3.6.1. Create an automatic snapshot policy

Automatic snapshot policies can be applied to system disks and data disks to periodically create snapshots for the disks. You can use automatic snapshot policies to improve data security and fault tolerance.

Background information

Automatic snapshot policies can effectively eliminate the following risks that may be brought about by manual snapshots:

- You may deploy applications such as personal websites or databases on an ECS instance. If the instance is exposed to attacks or encounters system vulnerabilities, you may be unable to manually create snapshots at the earliest opportunity. In this case, you can use the latest automatic snapshots to roll back the affected disks to restore data and reduce loss.
- You can also specify an automatic snapshot policy to create snapshots before you perform regular system maintenance tasks. This ensures that snapshots are created on a regular basis and eliminates the need to manually create snapshots.

You can retain up to 64 snapshots for each disk. If the maximum number of snapshots for a disk is reached, the earliest automatic snapshot is automatically deleted when a new automatic snapshot is created.

Procedure

- 1. Log on to the ECS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Automatic Snapshot Policies.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Click Create Policy.
- 5. In the **Create Policy** dialog box, set the parameters as required. The following table describes the parameters.

Parameter	Required	Description
Organization	Yes	The organization to which the automatic snapshot policy applies.
Resource Set	Yes	The resource set to which the automatic snapshot policy applies.
Region	Yes	The region to which the automatic snapshot policy applies.
Policy Name	Yes	The name of the automatic snapshot policy. The name must be 2 to 128 characters in length and can contain letters, digits, periods (.), underscores (_), hyphens (-), and colons (:). The name cannot start with a special character or a digit.
Sharing Scope	Yes	The scope in which the automatic snapshot policy can be shared. Valid values: Current Organization and Subordinate Organizations Current Resource Set Current Organization
	Yes	The points in time at which to create automatic snapshots. Valid values: 00:00 to 23:00 (the start of each hour). You can select multiple points of time. Onte The default time zone for the snapshot policy is UTC+8. You can change the time zone based on your business requirements.
Creation Time		If the amount of time scheduled for creating an automatic snapshot exceeds the time interval when a previous automatic snapshot is being created, the new snapshot creation task is skipped. This may occur when a disk contains a large volume of data. For example, you have an automatic snapshot policy that applies to a disk containing a large volume of data, and the policy specifies that snapshots are created at 00:00, 01:00, and 02:00. If the system starts to create a snapshot at 00:00 and spends 70 minutes processing the snapshot creation task, the system skips the automatic snapshot creation task scheduled at 01:00 and creates the next automatic snapshot at 02:00.
Frequency	Yes	The days of the week on which to create automatic snapshots. You can select one or more days from Monday to Sunday.
Retention Period	No	 The retention period of automatic snapshots. The default retention period is 30 days. The following options are available: Custom Period: You can select this option and then specify the number of days for which the created automatic snapshots are retained. Valid values: 1 to 65,535. Permanently: You can select this option to retain the created automatic snapshots for a longer period. After the maximum number of automatic snapshots for a disk is reached, the earliest snapshot is deleted when a new snapshot is created.

6. Click **OK**.

What to do next

After the automatic snapshot policy is created, you must apply the policy to a disk. For more information, see Enable or disable an automatic snapshot policy.

3.6.2. Search for and view an automatic snapshot policy

You can search for and view the created automatic snapshot policies.

Procedure

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Automatic Snapshot Policies.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. On the Automatic Snapshot Policies tab, use one of the following methods to search for the required automatic snapshot policy:

• Select a field from the field drop-down list and enter a keyword in the search box. Then, click the 🔍 icon. The automatic snapshot policies that

meet the filter condition are displayed in the list.

0	conditions	nced Filter . S are displayed i	pecify va n the list	lues for dif	ferent f	ields as r	required a	and clic	k Search	. The auto	matic sna	apshot polici	es that	meet the	e filter
I	0														

When you use the advanced intering reache, you can specify multiple inter conditions to narrow down search results.									
Field	Description								
Policy Name	Enter a complete policy name or a keyword.								
Policy ID	Enter a complete policy ID or a keyword.								
Tag	Select a tag key or a tag value to search for the automatic snapshot policies to which the tag is added.								

3.6.3. Modify an automatic snapshot policy

You can modify the configurations of a created automatic snapshot policy, including the policy name, snapshot creation time, execution frequency, and retention period.

Procedure

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Automatic Snapshot Policies.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find the automatic snapshot policy that you want to modify and click Modify Policy in the Actions column.
- In the Modify Policy dialog box, modify the configurations of the automatic snapshot policy. For more information about policy configurations, see Create an automatic snapshot policy.

⑦ Note Changes made to the retention period do not affect the existing snapshots and take effect only on subsequent snapshots.

6. Click **OK**.

3.6.4. Enable or disable an automatic snapshot policy

You can enable automatic snapshot policies for disks. After an automatic snapshot policy is enabled for a disk, snapshots are automatically created for the disk based on the time points and frequency specified in the policy.

Background information

We recommend that you enable automatic snapshot policies to create automatic snapshots during off-peak hours. You can manually create snapshots for disks that already have automatic snapshot policies enabled. When an automatic snapshot is being created for a disk, you must wait for the creation task to be complete before you can create a manual snapshot for the disk.

Procedure

- 1. Log on to the ECS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Automatic Snapshot Policies.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find the automatic snapshot policy that you want to enable or disable and click Apply or Cancel Policy in the Actions column.
- 5. In the Apply or Cancel Policy dialog box, enable or disable the automatic snapshot policy for disks based on your business requirements.
 - If you want to enable the automatic snapshot policy, click the Disks Without Policy Applied tab, select disks as required, and then click Apply Policy in the lower part of the disk list.
 - If you want to disable the automatic snapshot policy, click the Disks with Policy Applied tab, select disks as required, and then click Cancel Policy in the lower part of the disk list.

3.6.5. Delete an automatic snapshot policy

You can delete automatic snapshot policies that are no longer needed. After you delete an automatic snapshot policy, the policy is automatically disabled for the disks that have the policy enabled.

Procedure

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Storage & Snapshots > Automatic Snapshot Policies.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Find the automatic snapshot policy that you want to delete and click Delete Policy in the Actions column.
- 5. In the message that appears, click **Delete**.

3.7. Async replication

3.7.1. Create a replication pair

Before you can use the async replication feature to implement disaster recovery across zones within the same region, you must create a replication pair. This topic describes how to create a replication pair.

Prerequisites
The primary disk from which to replicate data and the secondary disk to which to replicate data are created. When you create a secondary disk, make sure that the secondary disk and the primary disk are of the same disk category and have the same performance level and capacity. For more information, see Create a disk. The primary and secondary disks cannot be located in the same zone.

Background information

When you create a replication pair, take note of the following items:

- You can select a premium performance disk or system disk as the primary disk in the replication pair. The selected disk cannot be encrypted or being resized.
- The async replication feature replicates data from primary disks to secondary disks and the replicated data overwrites the original data on the secondary disks. We recommend that you use an empty disk as the secondary disk. The primary and secondary disks must be of the same category and have the same performance level and capacity.
- A disk can belong to only one replication pair.

Procedure

- 1. Log on to the EBS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Block Storage.
- 2. In the left-side navigation pane, choose Enterprise Feature > EBS Async Replication.
- 3. In the top navigation bar, select an organization and a region.
- You need to select the region in which the disk that serves as the primary disk in the replicate pair is located.
- 4. On the EBS Async Replication page, click Create EAR.
- 5. In the Create Replication Pair panel, set the parameters as required and click Confirm. The following table describes the parameters.

Parameter	Description	
Region	The region in which the primary disk is located.	
Zone	The zone in which the primary disk is deployed.	
Disk ID	The ID of the primary disk.	
Region copied	The region in which the secondary disk is located.	
Zones of	The zone in which the secondary disk is deployed.	
Copy Disk ID	The ID of the secondary disk.	
Copy pair name	 The custom name that is used to identify the replication pair. The name must meet the following requirements: The name must be 2 to 128 characters in length. It must start with a letter and cannot start with http:// or <a href="http://. It can contain letters, digits, colons (:), underscores (_), periods (.), and hyphens (-). 	
Description	The description of the replication pair.	
RPO	The recovery point objective (RPO) of the replication pair. Valid values: 2 to 10080. Unit: minutes.	
Replication bandwidth	The bandwidth value used for the async replication. The greater the bandwidth value, the higher the rate at which data is asynchronously replicated. Unit: Mbit/s. Minimum value: 80. Maximum value: 640. Step size: 8.	

What to do next

After the replication pair is created, you must enable the async replication feature to implement disaster recovery for disks across zones in the same region. For more information, see Activate a replication pair to enable async replication.

If you want to manage multiple replication pairs with the same data replication direction, you can create a replication pair-consistent group and add the replication pairs to the replication pair-consistent group. For more information, see Create a replication pair-consistent group and Add replication pairs to a replication pair-consistent group.

3.7.2. View replication pairs

This topic describes how to view the created replication pairs.

Procedure

- 1. Log on to the EBS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > Elastic Block Storage.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Asynchronous Replication.
- 3. In the top navigation bar, select an organization and a region.
- 4. On the **EBS Async Replication** page, enter the IDs of one or more replication pairs in the search box next to the **PairID** drop-down list. To search for multiple replication pairs, separate their IDs with commas (,) or spaces. Then, click the click the

displayed in the replication pair list.

5. On the Primary Site or Secondary Site tab, view the details of each replication pair from multiple columns. The columns include PairID/Name, Description, Status, Primary Disk/Region/Zone, Secondary Disk/Region/Zone, Replication Group, Creation Time, Bandwidth(Mbps), and Recent recovery point.

② Note If the statuses of a replication pair on the Primary Site and Secondary Site tabs are different for an extended period of time, contact Alibaba Cloud technical support.

3.7.3. Modify a replication pair

After you create a replication pair, you can modify the name, RPO, and bandwidth of the replication pair based on your business requirements.

Background information

When you modify a replication pair, take note of the following items:

- You can modify only a replication pair that is in the Created or Stopped state.
- If a replication pair is added to a replication pair-consistent group, the RPO of the replication pair cannot be modified.

Procedure

1. Log on to the EBS console.

- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > Elastic Block Storage.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Asynchronous Replication.
- 3. In the top navigation bar, select an organization and a region.
- You need to select the region in which the primary disk in the replicate pair is located.
- 4. On the EBS Async Replication page, click the Primary Site tab.
- 5. Find the replication pair that you want to modify and click **Modify** in the **Operation** column.
- In the Modify Replication Pair dialog box, modify the name, description, RPO, and bandwidth of the replication pair as required, and then click Confirm.

For more information about the parameters used to configure a replication pair, see Create a replication pair. If the RPO and bandwidth of the replication pair are modified, the modifications take effect when data is synchronized in the next RPO period.

3.7.4. Activate a replication pair to enable async replication

A replication pair consists of a primary disk and a secondary disk that reside in different zones of the same region. To ensure that data can be asynchronously replicated from the primary disk to the secondary disk on a periodic basis, you must activate the replication pair to enable the async replication feature. This topic describes how to activate a replication pair to enable the async replication feature.

Prerequisites

A replication pair is created. For more information, see Create a replication pair.

Background information

When you activate a replication pair to enable the async replication feature, take note of the following items:

- If you require automatic synchronization, make sure that the replication pair is in the Created, Initial Syncing, Syncing, Normal, or Stopped state.
- If require manual synchronization, make sure that the replication pair is in the Created, One-time Synchronizing, or Stopped state.
- If you have added the replication pair to a replication pair-consistent group, you cannot separately perform activate, stop, failover, and reverse replication operations on the replication pair.

Procedure

- 1. Log on to the EBS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > Elastic Block Storage.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Asynchronous Replication.
- 3. In the top navigation bar, select an organization and a region.
- You need to select the region in which the primary disk in the replicate pair is located.
- 4. On the EBS Async Replication page, click the Primary Site tab
- 5. In the replication pair list, find the created replication pair that you want to activate and click Activate in the Operation column.

(2) Note If you want to activate multiple replication pairs at a time, select the replication pairs and click Batch Activate in the lower part of the page.

- 6. Create a snapshot for the secondary disk. This step is required if you want to protect data on the secondary disk. After async replication is enabled for a replication pair, data stored on the secondary disk is overwritten by data synchronized from the primary disk. To prevent loss of data on the secondary disk, we recommend that you create a snapshot for the secondary disk.
 - i. In the Activate Replication Pair dialog box, click Create Snapshot.
- ii. In the Create Snapshot dialog box, enter a snapshot name and set tags for the snapshot based on your business requirements. Then, click Create.
- 7. In the Activate Replication Pair dialog box, click Copy Data or OK based on your business requirements.
 - If you click **Copy Data**, the system immediately starts data synchronization.
 - If you click OK, the system starts a synchronization task and initiates data synchronization after half of the recovery point objective (RPO) period.

If you activate a replication pair for the first time, the replication pair enters the **Initial Syncing** state. You need to wait until data is synchronized from the primary disk to the secondary disk.

After the replication pair is activated and initial data synchronization is complete, the state of the replication pair changes to **Normal**. Subsequently, the system asynchronously replicates data from the primary disk to the secondary disk based on the RPO of the replication pair. This implements disaster recovery across zones in the same region. In the **Recent recovery point** column that corresponds to the replication pair, you can check when data was last replicated from the primary disk to the secondary disk.

3.7.5. Stop a replication pair to disable async replication

After you activate a replication pair to enable the async replication feature, you can stop the replication pair to disable the feature if you no longer need data replication or want to perform a failover.

Background information

When you stop a replication pair to disable the async replication feature, take note of the following items:

- The replication pair must be in the Initial Syncing, One-time Synchronizing, Syncing, Normal, Stopping, Stopped, or Stop Failed state.
- If you have added the replication pair to a replication pair-consistent group, you cannot separately perform activate, stop, failover, and reverse replication operations on the replication pair.

Procedure

1. Log on to the EBS console.

- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > Elastic Block Storage.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Asynchronous Replication.
- 3. In the top navigation bar, select an organization and a region.
- You need to select the region in which the primary disk in the replicate pair is located.
- 4. On the EBS Async Replication page, click the Primary Site tab.
- 5. In the replication pair list, find the created replication pair that you want to stop and click **Stop** in the **Operation** column.

() Note If you want to stop multiple replication pairs at a time, you can select the replication pairs and click **Batch Stop** in the lower part of the page.

The state of the replication pair changes to Stopped.

3.7.6. Use the asynchronous replication feature to implement disaster

recovery

After you create and activate a replication pair, if the primary disk in the replication pair fails, you can use the asynchronous replication feature to implement disaster recovery for the primary disk. This topic describes how to use the asynchronous replication feature to implement disaster recovery.

Background information

The asynchronous replication feature provides the failover and reverse replication sub-features. If the primary disk in a replication pair fails, you can use the failover sub-feature to enable read and write permissions on the secondary disk, attach the secondary disk to a temporary Elastic Compute Service (ECS) instance, and then fail over to the secondary disk. This allows you to continue your business. After the primary disk recovers, you can use the reverse replication sub-feature to replicate the latest data from the secondary disk to the primary disk for disaster recovery.

If you have added the replication pair to a replication pair-consistent group, you cannot separately perform activate, stop, failover, and reverse replication operations on the replication pair.

Step 1: Perform a failover

We recommend that you create a temporary Elastic Compute Service (ECS) instance in advance within the region and zone in which the secondary disk resides. This way, if the primary disk fails, you can use the failover sub-feature to enable read and write permissions on the secondary disk, attach the secondary disk to the temporary ECS instance, and then fail over to the secondary disk. You can continue your business on the secondary disk until the primary disk recovers.

- 1. Log on to the EBS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > Elastic Block Storage.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Asynchronous Replication.
- 3. In the top navigation bar, select an organization and a region.
- You need to select the region in which the primary disk in the replicate pair is located.
- 4. On the EBS Async Replication page, click the Primary Site tab.
- 5. Find the replication pair to which the faulty primary disk belongs, move the pointer over the _____ lcon in the **Actions** column, and then click **Failover**.
- 6. In the Failover dialog box, read the notes and click OK.

Asynchronous replication is suspended during a failover. To prevent data loss, we recommend that you use the failover sub-feature only when your primary disk fails.

After the failover sub-feature is enabled, the state of the replication pair changes to **Failover Completed**. To continue your business, you can attach the secondary disk to a temporary ECS instance and fail over to the secondary disk.

? Note

- Before you cancel a failover, make sure that the primary disk is available. Otherwise, the failover may fail to be canceled. In addition, after the failover is canceled, the replication pair is stopped. You must manually activate the replication pair.
- After the failover is complete, move the pointer over the π icon, click **Cancel Failover**, and then click **OK**.

Step 2: Perform a reverse replication

After the primary disk recovers, you can use the reverse replication sub-feature to replicate the latest data from the secondary disk to the primary disk for disaster recovery.

? Note

Before you use this sub-feature to perform a reverse replication, make sure that the primary disk is detached from its associated ECS instance and is in the Unattached state. For more information, see Detach a data disk.

- 1. In the top navigation bar, select an organization and a region.
- You need to select the region in which the primary disk in the replicate pair is located.
- 2. On the EBS Async Replication page, click the Primary Site tab.
- 3. Find the replication pair on which the failover is complete, move the pointer over the _____ icon in the Actions column, and then click Reverse

Replication.

4. In the Reverse Replication dialog box, read the notes and click Create Snapshot to create a snapshot for the primary disk.

In a reverse replication, the original data stored on the primary disk is overwritten by the data replicated from the secondary disk. To prevent loss of data on the primary disk, we recommend that you create a snapshot for the primary disk before a reverse replication. If you have manually created a snapshot for the primary disk after the primary disk recovers, you do not need to create a snapshot for the primary disk in this dialog box.

5. After a snapshot is created for the primary disk, click **OK**.

The state of the replication pair changes to Stopped.

! Important

After the reverse replication, the replication relationship in the replication pair is reversed. The original primary disk automatically becomes the secondary disk, and the original secondary disk automatically becomes the primary disk. For example, before a reverse replication is performed for a replication pair, the primary disk is Disk C in Zone B of Region A, and the secondary disk is Disk E in Zone D of Region A. After the reverse replication, the primary and secondary roles in the replication pair are interchanged. Disk E in Zone D of Region A becomes the primary disk, and D of Region A becomes the secondary disk.

6. In the Actions column, click Activate.

You must activate the replication pair in this step to asynchronously replicate data from the original secondary disk to the original primary disk. After data is asynchronously replicated from the original secondary disk to the original primary disk, the state of the replication pair changes to **Normal** and disaster recovery is complete.

7. Optional. Revert the replication relationship between the primary disk and secondary disk in the replication pair.

After you perform the preceding steps to perform a reverse replication, the original replication relationship in the replication pair is reversed. If you want to revert the replication relationship, perform the following steps:

- i. View the region in the Secondary Disk/Region/Zone column of the replication pair and select the region in the top navigation bar to switch to that region.
- ii. Find the replication pair on which you have performed a reverse replication. Move the pointer over the icon in the Actions column and click

Failover and then Reverse Replication.

iii. After the replication relationship in the replication pair is reverted, click Activate in the Actions column to activate the replication pair again.

3.7.7. Delete a replication pair

The functionality of disks in a replication pair is limited. If an existing replication pair is no longer needed for disaster recovery or needs to be replaced, you can delete it. This topic describes how to delete a replication pair.

Background information

- When you delete a replication pair, take note of the following items:
- The replication pair must be in the Created, Creation Failed, Stopped, Failover Completed, Deleting, Deletion Failed, Deleted, or Invalid state.
- After the replication pair is deleted, the primary and secondary disks in the replication pair are retained.
- If the replication pair is added to a replication pair-consistent group, you can delete the replication pair only after it is removed from the replication pair-consistent group.
- After a replication pair is stopped and deleted, data is no longer replicated from the primary disk to the secondary disk on a periodic basis. Before
 you delete a replication pair, properly manage your business status and business data storage to ensure data security of the primary disk and
 prevent data loss. We recommend that you create disk snapshots or snapshot-consistent groups to back up disk data. For more information about
 snapshots, see Create a snapshot.

Procedure

- 1. Log on to the EBS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > Elastic Block Storage.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Asynchronous Replication.
- 3. In the top navigation bar, select an organization and a region.
- You need to select the region where the disk that serves as the primary disk in the replicate pair resides.
- 4. On the EBS Async Replication page, click the Primary Site tab.
- 5. In the replication pair list, find the created replication pair that you want to delete and click **Delete** in the **Operation** column.

6. In the message that appears, click OK. When the replication pair is deleted, the secondary disk rolls back to the point in time when the last async replication was complete and drops all the data that is being replicated from the primary disk. The functionality limits are lifted from the secondary disk. You can attach the secondary disk and read data from or write data to the secondary disk.

3.8. Storage sets

3.8.1. Create a storage set

You can classify Elastic Block Storage (EBS) clusters based on dimensions such as the business type, and associate the EBS clusters with different partitions in a storage set. This allows you to isolate EBS clusters of different business types.

Background information

When you create a storage set, take note of the following items:

- The number of partitions in a storage set must be greater than or equal to 2.
- If you want to create a disk in a specified EBS cluster, create a storage set, and associate the EBS cluster with a partition of the storage set on CDS
 Ops, which is the unified O&M platform of Cloud Defined Storage (CDS). Then, select the storage set and partition to which the EBS cluster belongs
 when you create a disk by using the storage set in the Elastic Compute Service (ECS) console. For more information about how to associate an EBS
 cluster with a partition in a storage set, see the "Manage storage sets" topic under Resource management in CDS O&M Guide.

Procedure

- 1. Log on to the EBS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > Elastic Block Storage
- 2. In the left-side navigation pane, choose Enterprise Feature > Storage Set.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. On the Storage Set page, click Create a Storage Set.
- 5. In the Create a Storage Set panel, set the parameters as described in the following table and click OK.

Parameter	Description	
Region	The region in which you want to create the storage set.	
Zone	The zone in which you want to create the storage set.	
Storage Set Name	The name of the storage set. It is the unique identifier of the storage set. The name must meet the following requirements: The name must be 2 to 128 characters in length. The name must start with a letter and cannot start with http://or http://. The name can contain digits, colons (:), underscores (_), and hyphens (-). 	
Partitions	The number of partitions in the storage set. The number of partitions cannot be less than 2.	

What to do next

After you create a storage set, you can click the ID of the storage set and create a disk in the storage set. For more information, see Create a disk.

3.8.2. View storage sets

This topic describes how to view the created storage sets and their details.

Procedure

- 1. Log on to the EBS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Storage > Elastic Block Storage**.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Storage Set.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. On the Storage Set page, select a filter condition, enter the corresponding information in the search box, and then click the condition. The one or

more storage sets that meet the filter condition are displayed in the storage set list.

Filter condition	Description
Storage Set ID	Enter the full ID of a storage set to search for the storage set. Only exact match is supported. Fuzzy match is not supported.
Storage Set Name	Enter the name or a name keyword of a storage set to search for the storage set. Fuzzy match is supported.

3.8.3. Modify a storage set

After you create a storage set, you can modify the name of the storage set based on your business requirements.

Procedure

- 1. Log on to the EBS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > Elastic Block Storage.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Storage Set.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. On the Storage Set page, find the storage set that you want to modify and click Modify in the Operation column.
- 5. In the Modify Storage Set dialog box, modify the name of the storage set and click Confirm.

3.8.4. Delete a storage set

This topic describes how to delete a storage set that is no longer needed.

Background information

- When you delete a storage set, take note of the following items:
- If a disk exists in the storage set to be deleted, the system reports an error.

Before you delete a storage set, you must view the existing disks in the storage set and release them by performing the following steps:
 On the Storage Set page of the Elastic Block Storage (EBS) console, click the ID of the storage set that you want to delete. The Disks page in the Elastic Compute Service (ECS) console appears. On the Disks page, you can view the disks in the storage set.
 For more information about how to release a disk, see Release a data disk.

Procedure

- 1. Log on to the EBS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Storage > Elastic Block Storage**.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Storage Set.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. On the Storage Set page, find the storage set that you want to delete and click Delete in the Operation column.
- 5. In the message that appears, click **OK**.

3.9. Replication pair-consistent groups

3.9.1. Create a replication pair-consistent group

You can create a replication pair-consistent group to manage replication pairs between the primary site and the secondary site in a centralized manner.

Background information

When you create a replication pair-consistent group, take note of the following items:

- Replication pair-consistent groups support asynchronous disaster recovery across regions or zones in the same region.
- If a replication pair and a replication pair-consistent group have the same primary region (production region), primary zone (production zone). secondary region (disaster recovery region), and secondary zone (disaster recovery zone), the replication pair and the replication pair consistent group replicate data in the same direction. A replication pair can be added to a replication pair-consistent group only if the replication pair and the replication pair-consistent group replicate data in the same direction.
- After replication pairs are added to a replication pair-consistent group, the recovery point objective (RPO) of the group takes effect on the replication pairs in place of their original RPOs.

Procedure

- 1. Log on to the EBS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Storage > Elastic Block Storage**.
- 2. In the left-side navigation pane, choose Enterprise Feature > Replication Pair-Consistent Group.
- 3. In the top navigation bar, select an organization and a region.
- You must switch to the production region of the replication pair-consistent group.
- 4. On the Replication Pair-Consistent Group page, click Create replication group.
- 5. On the Create Replication Pair-consistent Group page, set the parameters as described in the following table and click Confirm.

Parameter	Description	
Group Name	The name of the replication pair-consistent group. The name must be 2 to 128 characters in length. The name must start with a letter and cannot start with $http://$ or $https://$. The name can contain digits, colons (:), underscores (_), and hyphens (-).	
Production region	The region in which the primary site is deployed.	
Production Zone	The zone in which the primary site is deployed.	
Disaster Recovery Region	The region in which the secondary site is deployed.	
Disaster Recovery Zone	The zone in which the secondary site is deployed.	
RPO	The RPO of the replication pair-consistent group. Valid values: 2 to 10080. Unit: minutes.	

What to do next

After a replication pair-consistent group is created, you can add replication pairs that replicate data in the same direction as the group to the group. For more information, see Add replication pairs to a replication pair-consistent group

3.9.2. View replication pair-consistent groups

This topic describes how to view the created replication pair-consistent groups.

Procedure

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Replication Pair-Consistent Group.
- 3. In the top navigation bar, select an organization and a region.
- On the Replication Pair-Consistent Group page, enter the IDs of one or more replication pair-consistent groups in the search box next to the Replication Pair-consistent Group ID drop-down list. To search for multiple replication pair-consistent groups, separate their IDs with commas (,) or spaces. Then, click the Q icon. The replication pair-consistent groups with the specified IDs are displayed in the replication pair-consistent group

list.

5. On the **Primary Site** or **Secondary Site** tab, view the details of each replication pair-consistent group from multiple columns. The columns include Replication Group ID/name, Description, Status, Production Region/Zone, Disaster Recovery Region/Zone, Number of Replication Pairs, and Last Data Recovery Point.

Note If the statuses of a replication pair-consistent group on the Primary Site and Secondary Site tabs are different for an extended period of time, contact Alibaba Cloud technical support.

3.9.3. Modify a replication pair-consistent group

After a replication pair-consistent group is created, you can modify its name and recovery point objective (RPO) based on your business requirements.

Background information

Before you modify a replication pair-consistent group, make sure that the group is in the Created or Stopped state.

Procedure

- 1. Log on to the EBS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Storage > Elastic Block Storage**.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Asynchronous Replication.
- In the top navigation bar, select an organization and a region. You must switch to the production region of the replication pair-consistent group.
- 4. On the Replication Pair-Consistent Group page, click the Primary Site tab.
- 5. In the replication pair-consistent group list, find the replication pair-consistent group that you want to modify and click Modify in the Operation

column

6. In the Edit replication group dialog box, modify the parameters of the replication pair-consistent group as needed and click OK. For more information about the parameters, see Create a replication pair-consistent group.

For more information about the parameters, see Create a replication pair-consistent group. After you modify the RPO of a replication pair-consistent group, the new configuration takes effect when data is synchronized in the next RPO period.

3.9.4. Add replication pairs to a replication pair-consistent group

You can add replication pairs that replicate data in the same direction as a replication pair-consistent group to the group so that you can manage these replication pairs as a batch by using the group.

Prerequisites

- A replication pair-consistent group is created. For more information, see Create a replication pair-consistent group.
- Replication pairs are created. For more information, see Create a replication pair.

Background information

- When you add replication pairs to a replication pair-consistent group, take note of the following items:
- If a replication pair and a replication pair-consistent group have the same primary region (production region), primary zone (production zone), secondary region (disaster recovery region), and secondary zone (disaster recovery zone), the replication pair and the replication pair-consistent group replicate data in the same direction. A replication pair can be added to a replication pair-consistent group only if the replication pair and the replication pair-consistent group replicate data in the same direction.
- You can add up to 256 replication pairs to a single replication pair-consistent group.
- The replication pairs and the replication pair-consistent group must be in the Created or Stopped state.
- After replication pairs are added to a replication pair-consistent group, the recovery point objective (RPO) of the group takes effect on the replication pairs in place of their original RPOs.
- After you add a replication pair to a replication pair-consistent group, you cannot separately perform activate, stop, failover, and reverse replication operations on the replication pair. You cannot modify the recovery point objective (RPO) of the replication pair.

Procedure

- 1. Log on to the EBS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > Elastic Block Storage.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Replication Pair-Consistent Group.
- 3. In the top navigation bar, select an organization and a region.
- You must switch to the production region of the replication pair-consistent group.
- 4. On the Replication Pair-Consistent Group page, click the Primary Site tab.
- 5. Click the ID of the replication pair-consistent group that you want to manage.

6. On the details page of the replication pair-consistent group, click the Primary Site tab, and then click Add replication pair.

③ Note After replication pairs are added to the replication pair-consistent group, you can click Activate in the upper-right corner of the group details page to enable async replication for the replication pairs.

7. In the Add a disaster recovery pair to the consistency replication group dialog box, select the replication pairs that you want to add and click OK.

What to do next

After replication pairs are added to the replication pair-consistent group, you must enable async replication for the replication pairs in the replication pair-consistent group. For more information, see Activate a replication pair-consistent group to enable async replication.

3.9.5. Remove replication pairs from a replication pair-consistent group

You can remove multiple replication pairs from a replication pair-consistent group at a time. When a replication pair is removed from a replication pair consistent group, the replication pair is disassociated from the group but is not deleted.

Background information

Before you remove replication pairs from a replication pair-consistent group, make sure that the replication pair-consistent group is in the **Created** or **Stopped** state.

Procedure

- 1. Log on to the EBS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > Elastic Block Storage.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Replication Pair-Consistent Group.
- 3. In the top navigation bar, select an organization and a region.
- You must switch to the production region of the replication pair-consistent group.
- 4. On the Replication Pair-Consistent Group page, click the Primary Site tab.
- 5. Click the ID of the replication pair-consistent group that you want to manage.
- 6. On the details page of the replication pair-consistent group, click the Primary Site tab.

7. In the replication pair list, select the replication pairs that you want to remove. In the lower part of the page, click Batch Remove.

3.9.6. Activate a replication pair-consistent group to enable async

replication

After you add replication pairs to a replication pair-consistent group, you must activate the group to enable async replication for the replication pairs so that data can be asynchronously replicated from disks in the primary site to disks in the secondary site on a periodic basis. After the replication pair-consistent group is activated, the system synchronizes all data from disks in the primary site to disks in the secondary site. Subsequently, the system periodically synchronizes incremental data based on the recovery point objective (RPO) of the replication pair-consistent group.

Background information

• If you require automatic synchronization, make sure that the replication pair-consistent group is in the Created, Synchronizing, Normal, or

Stopped state.

• If you require manual synchronization, make sure that the replication pair-consistent group is in the **Created**, **One-time Synchronizing**, or **Stopped** state.

Procedure

- 1. Log on to the EBS console.
 - i. Log on to the Apsara Uni-manager Management Console.
 - ii. In the top navigation bar, choose **Products > Storage > Elastic Block Storage**.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Replication Pair-Consistent Group.
- 3. In the top navigation bar, select an organization and a region.
- You must switch to the production region of the replication pair-consistent group.
- 4. On the **Replication Pair-Consistent Group** page, click the **Primary Site** tab.
- 5. In the replication pair-consistent group list, find the created replication pair-consistent group that you want to activate and click **Activate** in the **Operation** column.

⑦ Note If you want to activate multiple replication pair-consistent groups at a time, select the replication pair-consistent groups and click Batch Activate in the lower part of the page.

- 6. In the Activate Replication Group dialog box, click Copy Data or OK based on your business requirements.
 - If you click Copy Data, the system immediately starts data synchronization.
- If you click **OK**, the system starts a synchronization task and initiates data synchronization after half of the RPO period.

If you activate a replication pair-consistent group for the first time, the state of the group changes to **Initial synchronization**. You must wait until all data is synchronized.

After the replication pair-consistent group is activated and initial data synchronization is complete, the state of the replication pair-consistent group changes to **Normal**. Subsequently, the system asynchronously replicates incremental data from disks in the primary site to disks in the secondary site based on the RPO of the replication pair-consistent group. This implements disaster recovery across zones in the same region. In the **Recent recovery point** column that corresponds to the replication pair-consistent group, you can check when data was last replicated from disks in the primary site to disks in the secondary site.

3.9.7. Stop a replication pair-consistent group to disable async replication

After replication pairs in replication pair-consistent groups are activated, if you no longer require data replication or you want to perform a failover, you can stop the replication pair-consistent groups to disable the async replication feature for the replication pairs.

Background information

Before you stop a replication pair-consistent group, make sure that the group is in the **One-time Synchronizing**, **Synchronizing**, **Normal**, **Stopping**, **Stop Failed**, or **Stopped** state.

Procedure

- 1. Log on to the EBS console.
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Storage > Elastic Block Storage**.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Asynchronous Replication.
- 3. In the top navigation bar, select an organization and a region.
- You must switch to the production region of the replication pair-consistent group.
- 4. On the Replication Pair-Consistent Group page, click the Primary Site tab.
- 5. In the replication pair-consistent group list, find the replication pair-consistent group that you want to stop and click Stop in the Operation column.

② Note If you want to stop multiple replication pair-consistent groups at a time, select the replication pair-consistent groups and click **Batch** stop in the lower part of the page.

The state of the replication pair-consistent group changes to Stopped.

3.9.8. Use replication pair-consistent groups to implement disaster

recovery

After you create and activate a replication pair-consistent group, if the disks in the primary site (primary disks) fail, you can use the replication pairconsistent group to implement disaster recovery for the primary disks. This topic describes how to use replication pair-consistent groups to implement disaster recovery for the primary disks.

Background information

Replication pair-consistent groups support the failover and reverse replication sub-features. When the primary disks fail, you can use the failover subfeature to enable read and write permissions on data stored on the disks in the secondary site (secondary disks) and replicate the data from the primary disks to the secondary disks. Then, you can attach the secondary disks to temporary Elastic Compute Service (ECS) instances that are created for the failover to continue your business. After the primary disks recover, you can use the reverse replication sub-feature to replicate the latest data stored on the secondary disks to the primary disks. This implements disaster recovery.

Step 1: Perform a failover

We recommend that you create temporary ECS instances in advance within the region and zone in which the secondary disks reside. This way, if the primary disks fail, you can use the failover sub-feature to enable read and write permissions on the secondary disks, attach the secondary disks to the temporary ECS instances, and then fail over to the secondary disks. You can continue your business on the secondary disks until the primary disks recover.

1. Log on to the EBS console.

- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > Elastic Block Storage.
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Asynchronous Replication.
- 3. In the top navigation bar, select an organization and a region.
- You must switch to the production region of the replication pair-consistent group.

- 4. On the Replication Pair-Consistent Group page, click the Primary Site tab
- 5. Find the replication pair-consistent group to which the failed primary disks belong, move the pointer over the ______ icon in the **Operation** column, and

then click Failover.

6. In the Failover dialog box, read the notes and click OK.

Async replication is suspended during a failover. To prevent data loss, we recommend that you use the failover sub-feature only when your primary disks fail

After the failover sub-feature is enabled, the state of the replication pair-consistent group changes to **Failover completed**. To continue your business, you can attach the secondary disks to the temporary ECS instances and fail over to the secondary disks.

Step 2: Perform a reverse replication

After the primary disks recover, you can use the reverse replication sub-feature to replicate the latest data stored on the secondary disks to the primary disks to implement disaster recovery

() Note Before you perform a reverse replication, make sure that the primary disks are detached from the corresponding ECS instances and are in the Unattached state. For more information, see Detach a data disk.

1. In the top navigation bar, select an organization and a region.

You must switch to the production region of the replication pair-consistent group.

- 2. On the Replication Pair-Consistent Group page, click the Primary Site tab.
- 3. Find the replication pair-consistent group on which you have performed a failover, move the pointer over the ______ icon in the **Operation** column, and

then click Reverse replication

- 4. In the **Reverse replication** dialog box, read the notes and click **Create Snapshot** to create snapshots for the primary disks. After a reverse replication is performed, the original data stored on the primary disks is overwritten by the data replicated from the secondary disks To prevent loss of data on the primary disks, we recommend that you create snapshots for the primary disks. If you have manually created snapshots for the primary disks after the primary disks recover, you do not need to create snapshots for the primary disks.
- After you create snapshots for the primary disks, click OK. The state of the replication pair-consistent group changes to Stopped.
- In the Operation column, click Activate. After this step is performed, the data stored on the original secondary disks is asynchronously replicated to the original primary disks.
- If the data on the original secondary disks is replicated to the original primary disks, Normal is displayed in the Status column that corresponds to the replication pair-consistent group, and disaster recovery is complete. 7. Optional: Revert the replication relationship between the primary and secondary sites in the replication pair-consistent group.
- After you perform the preceding steps to perform a reverse replication, the replication relationship between the original primary and secondary sites in the replication pair-consistent group is reversed. To revert the replication relationship, perform the following steps:
- View the region in the **Disaster Recovery Region/Zone** column that corresponds to the replication pair-consistent group and select the region in the top navigation bar to switch to that region.

column and perform a failover and reverse replication again.

iii. After the replication relationship in the replication pair-consistent group is reverted, click Activate in the Operation column to activate the replication pair-consistent group again.

3.9.9. Delete a replication pair-consistent group

You can delete a replication pair-consistent group that you no longer need.

Background information

- When you delete a replication pair-consistent group, take note of the following items:
- The replication pair-consistent group must be in the Created, Creation Failed, Stopped, Failover completed, Deleting, Deletion Failed, or Invalid state
- · Before you delete a replication pair-consistent group, make sure that all replication pairs in the replication pair-consistent group are removed. For more information, see Remove replication pairs from a replication pair-

Procedure

- 1. Log on to the EBS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > Elastic Block Storage
- 2. In the left-side navigation pane, choose Enterprise-Grade Capabilities > Replication Pair-Consistent Group.
- 3. In the top navigation bar, select an organization and a region.
- You must switch to the production region of the replication pair-consistent group.
- 4. On the Replication Pair-Consistent Group page, click the Primary Site tab.
- 5. Find the replication pair-consistent group that you want to delete and click Delete in the Operation column.

6. In the message that appears, click **OK**. After the replication pair-consistent group is deleted, the secondary disks roll back to the point in time when the last asynchronous replication was complete and drop all the data that is being replicated from the primary disks.

3.10. Appendix

3.10.1. Manage tags

You can classify and manage resources such as disks, snapshots, and automatic snapshot policies by adding tags to the resources. You can also filter the resources based on the tags

Background information

Each tag is defined by using a key-value pair. You can use tags to classify and manage resources from different dimensions.

- You can add up to 20 tags to a resource.
- A tag key can be up to 128 characters in length and cannot contain http:// or https:// . It cannot start with aligun or acs: . The key

cannot be empty

- A tag value can be up to 128 characters in length and cannot contain http:// or https:// . It cannot start with acs: . The value cannot be empty.
- The key and value of a tag must be encoded in the UTF-8 format.

Add one or more tags to multiple resources at a time

1. Log on to the ECS console.

- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose cascading menu items based on the type of resources to which you want to add tags.
 - If you want to add tags to disks, choose Storage & Snapshots > Disks.
 - If you want to add tags to snapshots, choose Storage & Snapshots > Snapshots and click the Disk Snapshots tab.
 - If you want to add tags to automatic snapshot policies, choose Storage & Snapshots > Automatic Snapshot Policies.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Select the resources to which you want to add tags. In the lower part of the page, click Add Tags.
- 5. In the Add Tags dialog box, add one or more tags to the selected resources.

O Note After you click Tags, you are redirected to the Tag management page. On this page, you can create tags and add them to resources, view the resources to which the tags are added, and remove the tags from the resources.

- i. Click Add and enter the tag key and tag value.
- ii. Click OK
- iii. In the The tags are edited for the resources, message, view the selected resources and the tags that are added to the resources, and then click Clos The added tags are displayed in the **Tag** drop-down list.

Use tags to filter resources

1. Log on to the ECS console.

- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose cascading menu items based on the type of resources to which you want to add tags.
 - If you want to add tags to disks, choose Storage & Snapshots > Disks.
 - If you want to add tags to snapshots, choose Storage & Snapshots > Snapshots and click the Disk Snapshots tab.
 - If you want to add tags to automatic snapshot policies, choose Storage & Snapshots > Automatic Snapshot Policies.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Use one of the following methods to search for resources that meet the specified condition.
 - In the filter list, set the filter option to Tag and select a tag key or tag value in the search box. The resources that meet the specified condition are displayed in the resource list
 - Click Advanced Filter, select a tag key or a tag value from the Tag drop-down list, and then click Search. The resources that meet the specified condition are displayed in the resource list.

Remove one or more tags from multiple resources at a time

- 1. Log on to the ECS console.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Elastic Computing > Elastic Compute Service.
- 2. In the left-side navigation pane, choose cascading menu items based on the type of resources to which you want to add tags.
- If you want to add tags to disks, choose Storage & Snapshots > Disks.
- If you want to add tags to snapshots, choose Storage & Snapshots > Snapshots and click the Disk Snapshots tab.
- If you want to add tags to automatic snapshot policies, choose Storage & Snapshots > Automatic Snapshot Policies.
- 3. In the top navigation bar, select an organization, a resource set, and a region.
- 4. Select the resources from which you want to remove tags. In the lower part of the page, click Remove Tags.
- 5. In the **Remove Tags** dialog box, remove one or more tags from the selected resources.

(?) Note After you click Tags, you are redirected to the Tag management page. On this page, you can create tags and add them to resources, view the resources to which the tags are added, and remove the tags from the resources

i. Click the icon next to a tag.

- To remove more tags, repeat this step multiple times.
- ii Click OK
- iii. In the The tags are edited for the resources, message, view the selected resources and the tags that are removed from the resources, and then click Clos The removed tags are no longer displayed in the **Tag** drop-down list.

4.Log Service 4.1. What is Log Service?

Log Service is a cloud-native monitoring and analysis platform that provides large-scale, low-cost, and real-time services to process logs, metrics, and traces. Log Service allows you to collect, transform, query, analyze, visualize, consume, and ship data. Log Service also allows you to configure alerts. This helps improve the digital capabilities of your business in scenarios such as R&D, O&M, operations, and security.

Log Service is tested and verified by Alibaba Group in various big data scenarios. You can collect, consume, query, and analyze data without the need to develop separate features.

Log Service provides the following features:

- Log collection: Log Service allows you to collect data by using multiple methods, such as Logtail and JavaScript. You can collect various types of data, including logs, metrics, and traces from multiple data sources. The data sources include Alibaba Cloud services, servers, applications, IoT devices, mobile devices, and open source software. You can also collect data that is transferred over standard protocols and data in multiple formats.
- Query and analysis: Log Service allows you to query and analyze the collected logs in real time and view analysis results on charts and dashboards.
 Alerting: Log Service can run query statements at regular intervals after an alert task is created. If the results match the specified conditions, Log
- Service sends an alert to the specified contacts in real time. You can specify the conditions and contacts when you create the alert task.
 Data transformation: Log Service provides more than 200 built-in functions, more than 400 regular expressions, and flexible user-defined functions
- Data transformation: Log Service provides more than 200 built-in functions, more than 400 regular expressions, and flexible user-defined functions that allow you to filter, split, convert, enrich, and replicate data. This feature meets your business requirements in multiple scenarios, such as data distribution, data standardization, and data integration.
- Real-time consumption: Log Service provides real-time consumption interfaces that log consumers can use to consume the collected logs.
- Shipping: Log Service allows you to ship the collected logs to Object Storage Service (OSS) in real time.

4.2. Quick start

4.2.1. Procedure

This topic describes the basic procedure to use Log Service. You can follow this procedure to create projects, create Logstores, and collect log data. The following figure shows the procedure.

Figure 1. Procedure



1. Optional. Obtain an AccessKey pair.

An AccessKey pair is a secure identity credential that you can use to call API operations and access your Alibaba Cloud resources. You can use the AccessKey pair to sign API requests and pass security authentication.

2. Create a project.

- Create a project in a specified region. For more information, see Create a project.
- 3. Create a Logstore.
- Create a Logstore for the project and specify the number of shards. For more information, see Create a Logstore.

4. Collect text logs.

Select a method to collect log data based on your business requirements. For more information, see Collect text logs.

- 5. Configure indexes, and query and analyze log data.
 - Before you can query and analyze log data in Log Service, you must enable the indexing feature and configure indexes. For more information, see Configure indexes.
 - After you enable the indexing feature and configure indexes, you can query and analyze log data in real time. Log Service allows you to query and analyze large amounts of log data in real time. For more information, see Log search overview and Log analysis overview.

- After you query and analyze log data, you can configure charts and dashboards to display query and analysis results. For more information, see Chart overview and Dashboard overview.
- 6. Configure alert rules.

Log Service allows you to configure alert rules based on query and analysis results. Log Service sends alert notifications based on the notification methods that you specify. For more information, see Configure an alert rule.

7. Consume logs

Log Service allows you to consume logs by using multiple methods, such as a Spark Streaming client, Storm spout, and Flink connector. For more information, see Real-time consumption.

4.2.2. Log on to the Log Service console

This topic describes how to log on to the Log Service console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel.
- A web browser is available. We recommend that you use Google Chrome.

Procedure

1. Enter the URL of the Apsara Uni-manager Management Console in the address bar and press the Enter key.

? Note

- $\circ~$ You can click the current language in the upper-right corner to switch to another language.
- 2. Enter your username and password.

Obtain the username and password from an operations administrator.

```
Note
```

• First logon

The first time that you log on to the Apsara Uni-manager Management Console, you need to change the password of your account. The password must be 10 to 32 characters in length. It must contain all of the following character types: uppercase letters, lowercase letters, digits, and special characters. Supported special characters include: ! @ # \$%

• Forget password

If you have forgotten your password, click Forgot Password. On the page that appears, enter the username of your account, the email address that was used to create the account, and the CAPTCHA code. Then, the system sends a link for resetting the password to the specified email address.

3. Click Log On.

- 4. In the top navigation bar, choose Products > Application Services > Log Service.
- 5. On the page that appears, select an organization and region, and then click Access as Administrator. The home page of the Log Service console is displayed.

4.2.3. Obtain an AccessKey pair

You can obtain an AccessKey pair in the Apsara Uni-manager Management Console.

Obtain the AccessKey pair of an organization

- 1. Log on to the Apsara Uni-manager Management Console.
- 2. In the top navigation bar, click Enterprise.
- 3. In the left-side navigation pane, choose **Resources > Organizations**.
- 4. In the organization list, click the name of the level-1 organization whose AccessKey pair you want to obtain.
- 5. On the right side of the page that appears, click Management AccessKey.
- 6. In the dialog box that appears, view the AccessKey pair of the organization.

Obtain the AccessKey pair of a personal account

- 1. Log on to the Apsara Uni-manager Management Console.
- 2. In the upper-right corner of the homepage, move the pointer over the profile picture and click User Information.
- 3. In the AccessKey Pair section, view the AccessKey pair of the personal account.

? Note

The AccessKey pair provides full access to cloud resources within your Alibaba Cloud account. You must keep the AccessKey pair confidential.

4.2.4. Manage a project

A project in Log Service is a resource management unit that is used to separate and manage different resources. This topic describes how to create and delete a project in the Log Service console.

Create a project

⑦ Note You can create up to 1,000 projects for each Apsara Stack tenant account.

- 1. Log on to the Log Service console
- 2. In the Projects section, click Create Project.

3. In the Create Project panel, configure the following parameters and click Create.

Parameter	Description
Project Name	The name of the project. The name must be unique in a region and cannot be changed after the project is created.
Project Description	The description of the project.
Region	Select a region based on log sources. After you create a project, you cannot change the region where the project resides or migrate the project to another region. If you want to collect logs from an Elastic Compute Service (ECS) instance, we recommend that you select the region where the ECS instance resides. This way, Log Service can collect logs over an internal network, which is more efficient.

View the endpoint of a project

After you create a project, you can view the endpoint of the project on the **Project Overview** page.

- 1. In the Projects section, click the name of the project.
- 2. On the **Project Overview** tab, view the endpoint of the project.

Delete a project

```
🔥 Warning
```

After you delete a project, all log data that is stored in the project and the configurations of the project are deleted and cannot be restored. Proceed with caution.

1. In the Projects section, find the project that you want to delete and click **Delete** in the Actions column.

2. In the **Delete Project** panel, select a reason in the Reason for Deletion section and click **OK**.

Project-related API operations

Action	Operation
Create a project	CreateProject
Delete a project	DeleteProject
Query a project	Query a specified project: GetProjectQuery all projects: ListProject
Modify a project	UpdateProject

For more information, see API Reference in Log Service Developer Guide.

4.2.5. Manage a Metricstore

This topic describes how to create, modify, and delete a Metricstore in the Simple Log Service console.

Prerequisites

A project is created. For more information, see Manage a project.

Create a Metricstore

- 1. Log on to the Simple Log Service console.
- 2. In the Projects section, click the project that you want to manage.
- 3. Choose **Time Series Storage > Metricstore**. On the Metricstore tab, click the **+** icon.
- 4. In the Create Metricstore panel, configure the parameters and click OK. The following table describes the parameters.

Parameter	Description		
Metricstore Name	The name of the Metricstore. The name must be unique in the project to which the Metricstore belongs. After the Metricstore is created, you cannot change the name of the Metricstore.		
	If you turn on Permanent Storage , Simple Log Service permanently stores the collected metrics in the Metricstore.		
Permanent Storage	Note If you query the data retention period by calling an SDK and the returned result is 3650, metrics are permanently stored.		

Data Retention Period	The retention period of the collected metric data in the Metricstore. Valid values: 15 to 3000. Unit: days. Metrics are automatically deleted after the specified retention period ends. You can configure the Data Retention Period parameter only if you turn off Permanent Storage . Note If you shorten the data retention period, Simple Log Service deletes all expired data within 1 hour. The data volume that is displayed for Storage Size(Log) on the homepage of the Simple Log Service console is updated on the next day. For example, if you change the value of the Data Retention Period parameter from 5 to 1, Simple Log Service deletes the metrics of the previous four days within 1 hour.		
Shards	The number of shards. Simple Log Service provides shards that allow you to read and write data. Each shard supports a write capacity of 5 MB/s and 500 writes/s and a read capacity of 10 MB/s and 100 reads/s. You can create up to 10 shards in each Metricstore. You can create up to 200 shards in each project. For more information, see Shard.		
Automatic Sharding	If you turn on Automatic Sharding , Simple Log Service increases the number of shards when the existing shards cannot accommodate the data that is written. For more information, see Manage shards.		
Enable Disaster Recovery	If you turn on Enable Disaster Recovery and both primary and secondary clusters are deployed for Simple Log Service, the logs in the Metricstore that is created are synchronized to the secondary cluster. This improves disaster recovery capabilities.		
Maximum Shards	If you turn on Automatic Sharding , you must configure this parameter to specify the maximum number of readwrite shards that can be created. Maximum value: 64.		

Modify the configurations of a Metricstore

1. On the Time Series Storage > Metricstore tab, find the Metricstore that you want to modify and choose Nodify.

2. In the upper-right corner of the Metricstore Attribute page, click Modify

- Modify the data retention period. For more information, see **Create a Metricstore**.
- Manage shards.

By default, two shards are created in a Metricstore. You can split a shard or merge shards based on your business requirements. For more information, see Manage shards.

3. Click **Save**.

Delete a Metricstore

- () Important
 - Before you delete a Metricstore, you must delete all Logtail configurations of the Metricstore.
 - If the data shipping feature is enabled for the Metricstore, we recommend that you stop writing data to the Metricstore and make sure that all data in the Metricstore is shipped before you delete the Metricstore.
 - If your Apsara Stack tenant account does not have permissions to delete a Metricstore, submit a ticket.

1. On the Time Series Storage > Metricstore tab, find the Metricstore that you want to delete and choose 🔛 > Delete.

🔥 Warning

After you delete a Metricstore, all metrics in the Metricstore are deleted and cannot be restored. Proceed with caution.

2. In the message that appears, click **OK**.

Delete metrics

After the retention period of metrics ends, the metrics are automatically deleted. To delete metrics, you can modify the **Data Retention Period** parameter.

() Important

If you shorten the data retention period, Simple Log Service deletes all expired data within 1 hour. The data volume that is displayed for **Storage Size(Log)** on the homepage of the Simple Log Service console is updated on the next day. For example, if you change the value of the Data Retention Period parameter from 5 to 1, Simple Log Service deletes the metrics of the previous four days within 1 hour.

4.2.6. Manage Logstores

A Logstore in Log Service is used to collect, store, and query logs. This topic describes how to create, modify, and delete a Logstore in the Log Service console.

Create a Logstore

NoteYou can create up to 200 Logstores in each project.

- 1. Log on to the Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. On the Log Storage > Logstores tab, click the \perp icon.

4. In the Create Logstore panel, configure the following parameters and click OK.

Parameter	Description		
Logstore Name	The name of the Logstore. The name must be unique in the project to which the Logstore belongs. After the Logstore is created, you cannot change the name of the Logstore.		
WebTracking	If you turn on WebTracking , you can collect data from HTML, HTML5, iOS, or Android platforms to Log Service by using the web tracking feature.		
	If you turn on Permanent Storage , Log Service permanently stores the collected logs.		
Permanent Storage	⑦ Note If you query the data retention period by calling an API operation and the returned result is 3650, the data is permanently stored.		
	If you do not turn on Permanent Storage , you must configure this parameter. The retention period of logs in the Logstore. Valid values: 1 to 3000. Unit: days. If logs are stored for a period that exceeds the value of this parameter, the logs are automatically deleted.		
Data Retention Period	Note If you shorten the data retention period, Log Service deletes all expired logs within 1 hour. The data volume that is displayed for Storage Size(Log) on the homepage of the Log Service console is updated the next day. For example, if you change the value of the Data Retention Period parameter from 5 to 1, Log Service deletes the logs of the previous four days within 1 hour.		
Shards	The number of shards. Log Service provides shards that allow you to read and write data. Each shard supports a write capacity of 5 MB/s and 500 writes/s and a read capacity of 10 MB/s and 100 reads/s. You can create up to 10 shards in each Logstore. You can create up to 200 shards in each project. For more information, see Log Service Product Introduction .		
Automatic Sharding	If you turn on Automatic Sharding , Log Service increases the number of shards when the existing shards cannot accommodate the data that is written.		
Enable Disaster Recovery	If you turn on Enable Disaster Recovery and both primary and secondary clusters are deployed for Log Service, the logs in the Logstore that is created are synchronized to the secondary cluster. This improves disaster recovery capabilities.		
Maximum Shards	If you turn on Automatic Sharding , you must configure this parameter to specify the maximum number of shards into which existing shards can be automatically split. Maximum value: 64.		
Log Public IP	If you turn on Log Public IP, Log Service adds the following information to the Tag field of the collected logs: •client_ip: the public IP address of the log source. •receive_time: the time at which Log Service receives the log. The value is a UNIX timestamp representing the number of seconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC.		

Modify the configurations of a Logstore

- 1. In the Projects section, click the project that you want to manage.
- 2. On the Log Storage > Logstores tab, find the Logstore whose configurations you want to modify, click the 📷 icon, and then select Modify.

3. On the Logstore Attributes page, click Modify.

- For more information about the parameters, see Create a Logstore.
- 4. Modify the configurations and click Save.

Delete a Logstore

If you no longer need a Logstore, you can delete the Logstore in the Log Service console.

() Important

- Before you can delete a Logstore, you must delete all Logtail configurations that are associated with the Logstore.
- If the log shipping feature is enabled for a Logstore, we recommend that you stop writing data to the Logstore before you delete the Logstore and make sure that all data in the Logstore is shipped.
- If you are not authorized to delete a Logstore by using your Apsara Stack tenant account, submit a ticket to delete the Logstore.

1. On the Log Storage > Logstores tab, find the Logstore that you want to delete, click the 📷 icon, and then select Delete.

\land Warning

After you delete a Logstore, all logs in the Logstore are deleted and cannot be restored. Proceed with caution.

2. In the **Delete** message, click **OK**.

Logstore-related API operations

Action	API operation
Create a Logstore	CreateLogstore
Delete a Logstore	DeleteLogstore
Query a Logstore	Query a Logstore: GetLogstoreQuery all Logstores: ListLogstore
Modify a Logstore	UpdateLogstore

4.2.7. Manage shards

Log data on which read and write operations can be performed is stored in a shard of a Logstore. This topic describes how to split, merge, and delete shards in the Log Service console.

Background information

When you create a Logstore, you must specify the number of shards for the Logstore. After the Logstore is created, you can split or merge shards to increase or decrease the number of shards in the Logstore.

- Each shard supports a write speed of up to 5 MB/s and a read speed of up to 10 MB/s. If the read speed or write speed of a shard cannot meet your business requirements, we recommend that you split the shard.
- You can split a shard of a Logstore on the Logstore Attributes page of the Logstore.
- If the data traffic is very small compared with the maximum read speed or write speed of a shard, we recommend that you merge the shard.

Split a shard

Each shard supports a write speed of up to 5 MB/s and a read speed of up to 10 MB/s. If the read speed or write speed of a shard cannot meet your business requirements, we recommend that you increase the number of shards. You can split a shard to increase the number of shards.

- 1. Log on to the Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. On the Log Storage > Logstores tab, find the Logstore that you want to manage, click the 📷 icon, and then select Modify.

4. On the Logstore Attributes page, click Modify.

5. Find the shard that you want to split and click **Split** in the Actions column.

⑦ Note				
You can split only a shard that is in the readwrite state.				
Total Shards:2 (Read/Write Instances:2, Read-only Instances:0)				
lq 1∫	Status ↓1 ℃	Beginkey/EndKey	Created At ↓↑	Actions
0	readwrite	00000000000000000000000000000000000000	2020-02-24 17:50:37	Split Merge
1	readwrite	80000000000000000000000000000000000000	2020-02-24 17:50:37	Split

- 6. Select the number of new shards that you want to generate after the original shard is split.
- 7. Click **OK**.
- 8. Click Save.

After you split the shard, the status of the shard changes from readwrite to readonly. You can continue to consume data from the shard that is in the readonly state. You cannot write data to this shard. New shards are in the readwrite state and are displayed below the original shard. The MD5 hash ranges of the new shards cover the MD5 hash range of the original shard.

Automatic sharding

Log Service provides the automatic sharding feature. After you enable the automatic sharding feature, a shard is automatically split if the following conditions are met:

- The data write speed exceeds the maximum write speed of the current shard for more than 5 minutes.
- The number of shards that are in the readwrite state does not exceed the maximum number of shards that is specified for the Logstore.

```
? Note
```

Automatic sharding is not performed on the shards that are split from a shard in the previous 15 minutes.

You can enable the automatic sharding feature when you create or modify a Logstore. If you turn on Automatic Sharding, you must configure the Maximum Shards parameter.

• Automatic Sharding

After you turn on Automatic Sharding, if the data write speed exceeds the maximum write speed of the current shard for more than 5 minutes, Log Service automatically splits the shard based on the data volume to increase the number of shards.

Maximum Shards

After you turn on Automatic Sharding, you must configure the Maximum Shards parameter to specify the number of new shards that you want to generate after a shard is automatically split. The maximum value is 64.

Merge shards

You can click Merge in the Actions column of a shard to merge shards. Log Service automatically locates a shard that is next to the specified shard and merges the two shards.

- 1. Log on to the Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. On the Log Storage > Logstores tab, find the Logstore that you want to manage, click the 📖 icon, and then select Modify.

4. On the Logstore Attributes page, click Modify.

5. Find the shard that you want to merge and click **Merge** in the Actions column.

Total Shards:2 (Read/Write Instances:2, Read-only Instances:0)				
bi 11	Status ↓ໂ ∏	Beginkey/EndKey	Created At ↓	Actions
0	readwrite	00000000000000000000000000000000000000	2020-02-24 17:50:37	Split Merge
1	readwrite	80000000000000000000000000000000000000	2020-02-24 17:50:37	Split

6. Click Save.

After you merge shards, the specified shard and the shard next to the specified shard are in the readonly state. A new shard is generated and is in the readwrite state. The MD5 hash range of the new shard covers the MD5 hash ranges of the original shards.

Delete a shard

Automatic deletion

If you specify a data retention period when you create a Logstore, the shards in the Logstore and the data stored in the shards are automatically deleted when the data retention period elapses.

Manual deletion

If you turn on Permanent Storage when you create a Logstore, we recommend that you delete the Logstore to delete the shards in the Logstore and the data stored in the shards. For more information, see Delete a Logstore.

Shard-related API operations

Action	API operation
Split a shard	SplitShard
Merge shards	MergeShards
Query shards	ListShard

4.2.8. Terms

4.2.8.1. Terms

This topic introduces the terms that are used in Log Service.

Basic resources

Term	Description
project	A project in Log Service is used to isolate the resources of different users and control access to specific resources.
Logstore	A Logstore in Log Service is used to collect, store, and query logs.
Metricstore	A Metricstore in Log Service is used to collect, store, and query metrics.
log	Logs are records of changes that occur in a system during the runtime of the system. The records contain information about the operations that are performed on specified objects and the results of the operations. The records are ordered by time.
log group	A log group is a collection of logs. A log group is the basic unit that is used to write and read logs. Logs in a log group contain the same metadata, such as the IP address and log source.
metric	Metrics are stored as time series.
shard	A shard is used to control the read and write capacities of a Logstore. In Log Service, data is stored in shards. Each shard has an MDS hash range, and each range is a left-closed, right-open interval. The ranges do not overlap with each other. Each range must be within the entire MDS hash range [000000000000000000000000000000000000

topic	A topic is a basic management unit in Log Service. You can specify topics when you collect logs. This way, Log Service can classify logs by topic.
endpoint	An endpoint of Log Service is a URL that is used to access a project and the data of the project. To access the projects in different regions, you must use different endpoints. To access the projects in the same region over an internal network or the Internet, you must also use different endpoints. For more information, see Obtain the endpoint of Log Service in Developer Guide.
AccessKey pair	An AccessKey pair is an identity credential that consists of an AccessKey ID and an AccessKey secret. The AccessKey ID and AccessKey secret are used for symmetric encryption and identity authentication. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt and verify a signature string. The AccessKey secret must be kept confidential. For more information, see Obtain an AccessKey pair in Developer Guide.

Data collection

Term	Description
Logtail	Logtail is used by Log Service to collect logs. For more information, see Collection by Logtail in the "Data collection" section.
Logtail configuration	A Logtail configuration is a set of policies that are used by Logtail to collect logs. The configuration includes the log source and collection method. For more information, see Collection by Logtail in the "Data collection" section.
machine group	A machine group is a virtual group that contains multiple servers. Log Service uses machine groups to manage the servers from which you want to collect logs by using Logtail. For more information, see Logtail machine group in the "Data collection" section.

Data query and analysis

Term	Description
query	You can specify filter conditions in search statements to obtain specific logs. For more information, see Log search overview in the "Query and analysis" section.
analysis	 You can invoke SQL functions on query results to perform statistical and analytical operations. Then, you can obtain analysis results. Log Service supports the SQL-92 syntax for log data analysis. For more information, seeLog analysis overview in the "Query and analysis" section. Log Service supports the SQL-92 syntax and the PromQL syntax for metric data analysis. For more information, see Query and analysis in the "Time series storage" section.
query statement	A query statement is in the Search statement Analytic statement format. A search statement can be executed alone. However, an analytic statement must be executed together with a search statement. The log analysis feature is used to analyze search results or all data in a Logstore. For more information, see Query and analysis .
index	 Indexes are a structure for storage. Indexes are used to sort one or more columns of data. You can query data only after you create indexes for the data. Log Service provides the following types of indexes: Full-text index: Log Service splits an entire log into multiple words based on specified delimiters to create indexes. In a search statement, the field names (keys) and field values (values) are plain text. Field index: After you configure field indexes, you can specify field names and field values in the Key:Value format to search for logs. For more information, see Configure indexes in the "Query and analysis" section.

Data consumption and shipping

Term	Description
consumer group	You can use consumer groups to consume data in Log Service. A consumer group consists of multiple consumers. Each consumer consumes different logs that are stored in a Logstore.
Alerting	

Term	Description
alert	An alert indicates an alert event. If an alert is triggered based on a specific alert monitoring rule, the alert management system sends the alert event to the notification management system. Log Service also provides alert-related subsystems, features, entities, and modules, such as the alert monitoring system and alert monitoring rules. For more information, see Alerts .

4.2.8.2. Log

Logs are records of changes that occur in a system during the running of the system. The records contain information about the operations that are performed on specific objects and the results of the operations. The records are ordered by time.

Format

Log data is stored in different formats, such as log files, log events, binary logs, and metric data. Log Service uses a semi-structured data model to define logs. A log consists of the following fields: topic, time, content, source, and tags. Log Service has different format requirements on different log fields. The following table describes the log fields and provides the format requirements.

Field	Description	Format
Торіс	The custom field in a log. This field can be used to identify the log topic. For example, you can set different log topics, such as access log and operation log, for website logs based on log types. For more information, see Topic .	The field value can be a string of up to 128 bytes. The value can include an empty string. If the field is an empty string, the log topic is not configured.
Time	The time when the log is generated, or the system time of the host where Logtail resides when the log data is collected. This field is a reserved field.	The value is a UNIX timestamp. It is the number of seconds that have elapsed since 00:00:00 UTC, Thursday, January 1, 1970.
Content	The content of the log. The content consists of one or more items. Each item is a key-value pair.	A key-value pair must comply with the following requirements: • The key is a UTF-8 encoded string of up to 128 bytes. The key can contain letters, digits, and underscores (_). The key cannot start with a digit. The string is 1 to 128 bytes in length. The following fields cannot be used: •time •source •topic •topic •extract_others_ •extract_others • The value can be a string of up to 1 MB.
Source	The source of the log. For example, the value of this field can be the IP address of the server where the log is generated.	The value of this field can be a string of up to 128 bytes.
Tags	 The tags of the log. Valid values: Custom tags: the tags that you add when you call the PutLogs operation to write logs to a specified Logstore. System tags: the tags that are added by Log Service. The tags include _client_ip_ and _receive_time 	The field value is in the dictionary format. The keys and the values are strings. The field name is prefixed by _tag_: .

Example

The following sample website access log shows the mapping between the raw log and the data model supported by Log Service.

127.0.0.1 - - [01/Mar/2021:12:36:49 0800] "GET /index.html HTTP/1.1" 200 612 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36

4.2.8.3. Log group

A log group is a collection of logs. A log group is the basic unit that is used to write and read logs. The logs in a log group contain the same metadata, such as the IP address and log source.

When you write logs to or read logs from Log Service, multiple logs are encapsulated into a log group. This way, you can write and read logs by log group. This method can reduce the number of read and write operations and improve business efficiency. The maximum length of a log group is 5 MB.

<pre>{Meta: {lp: 192.0.2.0 , Source: /home/admin/app.log,tag: az}, Logs: { { time:2020-05-05 19:27:28, user:1009, opt:pay, tranid:5618}, {time:2020-05-05 19:27:29, user:1003, opt:withdraw, tranid:561}</pre>	→ ((Meta • IP • Source • Tags	Logs • time, [<key,value>,] • time, [<key,value>,] • time, [<key,value>,]</key,value></key,value></key,value>
}}			

LogGroup

4.2.8.4. Project

A project in Log Service is used to isolate the resources of different users and control access to specific resources.

A project contains resources such as Logstores, Metricstores, and machine groups, and provides an endpoint that you can use to access the resources of Log Service. We recommend that you use different projects to manage the data in different applications, services, or projects.

You can use a project to organize and manage Logstores or Metricstores. You may need to use Log Service to collect and store the log data of
different projects, services, or environments. You can specify different projects to manage the log data. This facilitates the consumption, export, and
analysis of the log data.

- You can use a project to perform access control. You can grant the permissions to manage a specified project to a RAM user.
- A project provides an endpoint that you can use to access the resources in the project. Log Service allocates an exclusive endpoint to each project. You can use the endpoint of a project to read, write, and manage log data. For more information about the endpoints, see the "Obtain the endpoint of Log Service" topic in **Developer Guide**.

4.2.8.5. Logstore

A Logstore in Log Service is used to collect, store, and query log data.

Each Logstore belongs to a project. You can create multiple Logstores in a project. After you create a project, you can create multiple Logstores in the project to store different types of logs that are collected from the same application. For example, if you want to collect the operation logs, application logs, and access logs of App A, you can create a project named app-a and create three Logstores named operation_log, application_log, and access_log in the project to store the logs.

When you perform operations such as writing, querying, analyzing, consuming, or shipping logs, you must specify a Logstore.

- · Log Service uses Logstores as a collection unit to collect logs.
- Log Service uses Logstores as a storage unit to support operations such as storing, consuming, and shipping logs.
- Log Service creates indexes in a Logstore to support guery and analysis operations.

4.2.8.6. Metricstore

A Metricstore is a unit that is used to collect, store, and query metrics in Log Service.

Each Metricstore belongs to a project. You can create multiple Metricstores in a project. After you create a project, you can create multiple Metricstores in the project to store different types of metrics that are collected. For example, if you want to collect the metrics of hosts, cloud services, and business applications, you can create a project named demo-monitor and create three Metricstores named host-metrics, cloud-service-metrics, and app-metrics in the project to store the metrics.

You must specify a Metricstore when you write, query, analyze, or consume metrics.

- · Log Service uses a Metricstore as a collection unit to collect metrics.
- Log Service uses a Metricstore as a storage unit to store and consume metrics.
- Log Service supports the SQL-92 syntax and the PromQL syntax for you to guery and analyze metrics.

4.2.8.7. Metric

Log Service stores all data in Metricstores as time series. Log Service uses the model of time series data that is defined by Prometheus. Each time series consists of samples with the same metric identifier.

Metric identifier

Each time series has a unique metric identifier that consists of a metric name and a label.

Metric names are strings and must match the **[a-zA-Z_:][a-zA-ZO-9_:]*** regular expression. In most cases, a metric name indicates a description of a time series. For example, **http_request_total** indicates that each sample of a time series indicates the total number of received HTTP requests.

Labels are key-value pairs. Label keys must match the **[a-zA-Z_][a-zA-Z0-9_]*** regular expression. Label values can contain all characters except vertical bars ()). In most cases, a label indicates an attribute of a time series. For example, the value of the **method** key may be **POST**, and the value of the **URL** key may be **/api/v1/get**.

Samples

A sample indicates the value of a metric at a point in time. Each sample consists of a timestamp and a value. Timestamps are accurate to the nanosecond, and values are of the DOUBLE type.

Encoding format

When metric data is written to Log Service, the Protocol Buffer (Protobuf) format must be used. This format is also used to write log data. The metric identifier and samples are contained in the **content** field. The following table describes the related subfields.

Кеу	Description	Example
name	The name of the metric.	nginx_ingress_controller_response_size
labels	The labels of the metric. Format: {key}#\$#{value} {key}#\$#{value} {key}#\$#{value}.	app#\$#ingress- nginx controller_class#\$#nginx controller_namespace#\$#k ube-system controller_pod#\$#nginx-ingress-controller- 589877c6b7-hw9cj
time_nano	The timestamp of a sample. The value is accurate to the nanosecond.	1585727297293000000
value	The value of a sample.	36.0

4.2.8.8. Shard

Shards are used to manage the read and write capacity of Logstores or Metricstores. In Log Service, data is stored in a shard.

MD5 value range

- BeginKey: the start of a shard. The value is included in the MD5 value range of the shard.
- EndKey: the end of a shard. The value is excluded from the MD5 value range of the shard.

In this example, Logstore A has four shards. The following table describes the MD5 value range of each shard. Table 1. MD5 value range

Shard ID	Value range
Shard0	[00000000000000000000000000,40000000000
Shard1	[40000000000000000000000000000000000000
Shard2	[8000000000000000000000000,c000000000000
Shard3	[c00000000000000000000000000./ //////////

To read data from a shard, you must specify the ID of the shard. To write data to a shard, you can use the load balancing method or specify a hash key.

- If you use the load balancing method, each data packet is randomly written to an available shard.
- If you specify a hash key, data is written to the shard whose MD5 value range includes the value of the specified hash key.
- For example, you use the shard range that is shown in the preceding table. If you specify 5F as a hash key to write data to a Logstore, the data is written to Shard1 because the MD5 value range of Shard1 contains the hash key 5F. If you specify 8C as a hash key, the data is written to Shard2 because the MD5 value range of Shard2 contains the hash key 8C.

Shard capacity

Each shard provides the following read capacity and write capacity:

- Write capacity: 5 MB/s or 500 times/s
- Read capacity: 10 MB/s or 100 times/s

We recommend that you adjust the number of shards based on the actual data traffic. If the data traffic exceeds the read or write capacity of a shard, you can split the shard into multiple shards to increase the capacity. If the data traffic is much lower than the read or write capacity of a shard, you can merge the shard with another shard to reduce the capacity and save costs.

For example, you have two shards that are in the readwrite state and the shards can provide a write capacity of up to 10 MB/s. If you need to write data at a speed of 14 MB/s in real time, we recommend that you split one of your shards into two shards. This way, you can have three shards that are in the readwrite state. If you need to write data at a speed of 3 MB/s in real time, we recommend that you merge your shards.

- () Important
 - If the error code 403 or 500 is frequently returned when you write data by calling the Log Service API, you can go to the CloudMonitor console to check the traffic and status codes. Then, you can determine whether to increase the number of shards.
 - If the data traffic exceeds the capacity of your shards, Log Service attempts to provide services to meet your business requirements. However, Log Service cannot ensure the quality of the services.

Shard status

A shard can be in the readwrite state or readonly state.

When you create a shard, the shard is in the readwrite state. If you split a shard or merge shards, the status of the original shard changes to readonly. The newly generated shards are in the readwrite state. The status of a shard does not affect the read capacity of the shard. Data can be written to the shards that are in the readwrite state, but cannot be written to the shards that are in the readwrite state.

Splitting and merging

Log Service allows you to split and merge shards.

• After you split a shard, two more shards are added. The new shards are in the readwrite state and are listed under the original shard. The MD5 value range of the new shards includes the MD5 value range of the original shard.

You can split only a shard that is in the readwrite state. After you split a shard, the status of the shard changes from readwrite to readonly. This indicates that data can still be read from the shard, but cannot be written to the shard.

• You can merge two shards into one shard. The new shard is in the readwrite state and is listed under the original shards. The MD5 value range of the new shard includes the MD5 value range of the two original shards.

When you merge shards, you must specify a shard that is in the readwrite state. The shard cannot be the last shard in the shard list. Log Service finds the shard whose MD5 value range is next to the specified shard and then merges the two shards. After you merge the shards, the status of the shards changes from readwrite to readonly. This indicates that data can still be read from the shards, but cannot be written to the shards.

4.2.8.9. Topic

A topic is a basic management unit in Log Service. When you collect logs, you can specify topics to identify the logs.

You can use topics to identify logs that are generated by different services, users, and instances. For example, System A consists of an HTTP request processing module, a cache module, a logic processing module, and a storage module. You can set the log topic of the HTTP request processing module to http_module, the log topic of the cache module to cache_module, the log topic of the logic processing module to logic_module, and the log topic of the storage module to store_module. After the logs of the preceding modules are collected and saved to the same Logstore, you can identify logs based on the topics.

If you do not need to identify logs in a Logstore, set the topic to Null - Do not generate topic when you collect logs. A topic can be an empty string. The following figure shows the relationships between Logstores, topics, and shards.



4.2.9. Limits

4.2.9.1. Basic resources

This topic describes the limits on the basic resources of Simple Log Service.

Item	Description	Remarks
Project	You can create a maximum of 1000 projects within an Alibaba Cloud account.	If you want to increase the quota, contact Alibaba Cloud technical support.
Logstore	You can create a maximum of 800 Logstores in a project.	If you want to increase the quota, contact Alibaba Cloud technical support.
Shard	 You can create a maximum of 1,600 shards in a project. When you create a Logstore by using the console, you can create a maximum of 10 shards in the Logstore. When you create a Logstore by calling the API, you can create a maximum of 100 shards in the Logstore. You can split shards to increase the number of shards by using one of the preceding two methods. 	If you want to increase the quota, contact Alibaba Cloud technical support.
Logtail configuration	You can create a maximum of 800 Logtail configurations in a project.	If you want to increase the quota, contact Alibaba Cloud technical support.
Log retention period	You can permanently retain logs. You can also specify a log retention period. Unit: days. Valid values: 1 to 3000.	N/A
Machine group	You can create a maximum of 800 machine groups in a project.	If you want to increase the quota, contact Alibaba Cloud technical support.
Consumer group	You can create a maximum of 30 consumer groups in a Logstore.	You can delete consumer groups that are no longer used.
Saved search	You can create a maximum of 400 saved searches in a project.	N/A
Dashboard	You can create a maximum of 400 dashboards in a project.You can add a maximum of 800 charts to a dashboard.	N/A
LogItem	 The maximum size of a log that can be collected by calling the API is 10 MB. The maximum size of a log that can be collected by using Logtail is 512 KB. 	N/A
Field name (key)	The maximum size of a field name (key) is 128 bytes.	N/A
Field value (value)	The maximum size of a field value (value) is 1 MB.	N/A
Log group	The maximum size of a log group is 5 MB.	N/A

Alert rule

You can create a maximum of 400 alert rules in a project.

If you want to increase the quota, contact Alibaba Cloud technical support.

4.2.9.2. Data read and write

This topic describes the limits on data read and write in Simple Log Service.

Resource	Item	Description	Remarks
Project	Write traffic	The maximum write traffic of raw data is 120 GB per minute.	If the limit is exceeded, the HTTS status code 403 and the "Inflow Quota Exceed" error message are returned. If you want to increase the quota, contact Alibaba Cloud technical support.
	Number of write operations	The maximum number of write operations is 2,400,000 per minute.	If the limit is exceeded, the HTTS status code 403 and the "Inflow Quota Exceed" error message are returned. If you want to increase the quota, contact Alibaba Cloud technical support.
	Number of read operations	The maximum number of read operations is 2,400,000 per minute.	If the limit is exceeded, the HTTS status code 403 and the "Inflow Quota Exceed" error message are returned. If you want to increase the quota, contact Alibaba Cloud technical support.
	Write traffic	 If indexes are configured in a Logstore, the maximum write traffic of raw data is 5 MB per second. If no indexes are configured in a Logstore, the maximum write traffic of raw data is 10 MB per second. 	Not required. If the limit is exceeded, the system continues to provide the service. However, the quality of the service may be degraded.
Shard	Number of write operations	The maximum number of write operations is 500 per second.	Not required. If the limit is exceeded, the system continues to provide the service. However, the quality of the service may be degraded.
	Read traffic	The maximum read traffic is 10 MB per second.	Not required. If the limit is exceeded, the system continues to provide the service. However, the quality of the service may be degraded.
	Number of read operations	The maximum number of read operations is 100 per second.	Not required. If the limit is exceeded, the system continues to provide the service. However, the quality of the service may be degraded.

4.2.9.3. Logtail

This topic describes the limits of Logtail. Runtime environments

Item	Description
Architecture	 Linux Logtail supports x86_64 and ARM64. Windows Logtail supports x86_32 and x86_64.
Memory	 If no workloads are running and no plug-ins are enabled, a minimum of 20 MB of memory is required. If no workloads are running and at least one plug-in is enabled, a minimum of 120 MB of memory is required. The actual usage of memory varies based on the collection rate, monitored directories, number of log files, and number of synchronously sent requests.
Operating system	 Linux kernel 2.6.32 and later GNU C Library version 2.5 and later Windows Server 2004 and later
Kubernetes	 When you collect logs in DaemonSet mode, Kubernetes 1.10.0 or later is required. The HostToContainer mount propagation must be supported. When you use a custom resource definition (CRD) to collect logs, Kubernetes 1.16.0 or later is required, and the alibaba-log-controller component must be installed. The apiextensions.k8s.io/v1beta1 API provided by Kubernetes 1.7.0 and later also supports CRDs. However, the stability of the API in the Beta version varies based on the specified Kubernetes version.

Docker	The collection of stdout and stderr from containers has the following limits: • You must add "log-driver": "json-file" to the Docker configuration file daemon.json.
	• For CentOS 7.4 and later except Centos 8.0, you must set <pre>fs.may_detach_mounts</pre> to 1. For more information, seeBug 1468249, Bug 1441737, and Issue 34538.
	 logtail-ds: At least 0.1 cores and 256 MB of memory must be reserved for each node. alibaba-log-controller: At least 0.05 cores and 100 MB of memory must be reserved for each node.
The logtail-ds component for Container Service for Kubernetes (ACK)	Important The preceding components belong to the system-cluster-critical class. If cluster resources are insufficient, we recommend that you do not deploy the components. Otherwise, the existing pods on the node may be evicted.

File collection

Item	Description
Size of a single log	By default, the maximum size of a log is 512 KB. You can change the value of the startup parameter max_read_buffer_size to change the size. The maximum size of a log cannot exceed 8 MB. For more information, see Configure the startup parameters of Logtail If a multi-line log is split based on a regular expression that is used to match the beginning of the first line of a log, the maximum size of each log after splitting is still 512 KB. If the size of a log exceeds 512 KB, the log is forcefully split into multiple logs for collection. For example, if the size of a log is 1,025 KB, the log is split into logs of the following sizes: 512 KB, 512 KB, and 1 KB. Then, the logs are collected in sequence and considered incomplete logs.
Log file encoding	Logtail supports log files that are encoded in UTF-8 and GBK. We recommend that you use UTF-8-encoded log files to improve processing performance. A Warning If log files are encoded in other formats, issues such as garbled characters and data loss may occur.
Size of a log file	Unlimited.
Log file rotation	By default, the maximum number of log files in a rotation queue is 20. You can change the value of the startup parameters of Lograil. You can specify a log path in the xxx.log or xxx.log* format. Important Make sure that the two formats do not exist at the same time in a Logtail instance. If the two formats exist at the same time, the logs in a log file may be collected by using multiple Logtail configurations, and duplicate data may be collected. If more than 20 log files are not processed, new logs will be lost. In this case, you must check whether the write quota of shards exceeds the limit and adjust concurrency-related parameters. For more information, see Recommend parameter values.
Log collection behavior performed when log parsing is blocked	When log parsing is blocked, Logtail keeps the descriptor of the log file open to prevent the log file from being deleted during the blocking period and log loss. If the log file is rotated multiple times during the blocking period, Logtail puts the log file into a rotation queue.
Regular expressions	Logtail uses regular expressions that are compatible with Perl.
JSON	Standard JSON formats defined in RFC 7159 and ECMA-404 are supported. Non-standard JSON formats, such as { "name": "\xE5\xAD\xA6" }, are not supported.
File opening behavior	Logtail keeps the log files from which you want to collect logs and the log files in a rotation queue open to ensure the integrity of collected data. A log file is closed in the following scenarios: The log file is not modified within 5 minutes. The log file is rotated, and all logs in the log file are collected. The Logtail configuration is updated. If you want to release the file handle within a specified period of time after a log file is deleted, regardless of whether log collection from the log file is complete or whether new logs are still written to the log file, you can configure the force_release_deleted_file_fd_timeout parameter to specify a timeout period. For more information, seeConfigure the startup parameters of Logtail.
First log collection behavior	Logtail collects data only from incremental log files. If the size of a log file exceeds the limit of 1 MB the first time the modification to the log file is detected, Logtail collects data from the last 1 MB. If the size of the log file does not exceed 1 MB, Logtail collects data from the log file. The limit for container stdout and stderr is 512 KB. You can change the value of the tail_size_kb parameter in a Logtail configuration to change the limit.For more information, see Logtail configurations. If a log file is not modified after a Logtail configuration is delivered, Logtail does not collect data from the log file. For more information about how to collect historical log files, see Import historical log files.

File overwriting behavior	Logtail uses an inode and the hash value of the first 1,024 bytes of a log file to identify the log file. If a log file is overwritten and the inode or the hash value of the first 1,024 bytes of the log file changes, the log file is considered a log file from which logs are not collected, and the logs are collected from the beginning of the log file. If the inode or the hash value does not change, the logs in the log file are not collected.
File transfer behavior	If a log file is transferred and the matched Logtail configuration is not used to collect logs from the log file before the log file is transferred, the log file is considered a log file from which logs are not collected, and the logs are collected from the beginning of the log file. In this scenario, if the matched Logtail configuration is used to collect logs from the log file, the logs in the log file are not collected.
File collection history	 Logtail retains the historical collection progress of historical log files in the memory to ensure that only incremental data is collected after the log files are changed. If the historical collection progress of a historical log file is retained longer than a specified period and new data is written to the log file, duplicate data is collected. By default, the historical collection progress of historical log files is retained for up to one month. If the number of historical log files in the same directory exceeds 5,000, the historical collection progress of the log files within the previous week is retained. If the number of historical log files in the same directory exceeds 10,000, the historical collection progress of the log files within the previous day is retained. If you configure the Timeout parameter in a Logtail configuration, the historical collection progress of the log files within the last 30 minutes is retained.
Non-standard text logs	If a log contains \ 0 , the log is truncated at the first occurrence of\ 0 , and the rest of the log is discarded. For other escape characters such as ASCII codes that are used to represent colors or non-printable characters, Logtail directly sends the characters.

Container collection

? Note

Both the limits on log files and the limits on containers apply when you use Logtail to collect container logs.

Item	Description
Size of a single log	If you want to collect container stdout and stderr, the size of a log cannot exceed the threshold specified for log truncation. For logs of Alibaba Cloud services such as ACK and Elastic Container Instance (ECI), the threshold specified for log truncation is 16 KB.
First log collection behavior	When you collect container stdout and stderr and if the size of a log file exceeds the limit of 512 KB the first time the modification to the log file is detected, Logtail collects data from the last 512 KB. If the size of the log file does not exceed 512 KB, Logtail collects data from the beginning of the log file. You can change the value of the StartLogMaxOffset parameter in a Logtail configuration to change the limit. For more information, see Use the Simple Log Service console to collect container stdout and stderr in DaemonSet mode.
Symbolic link	When you collect logs from container files, files and directories of files cannot be symbolic links.
Container lifecycle	By default, Logtail can collect logs from a container only if the container lifecycle lasts 10 seconds or longer. When you collect logs from a container file, Logtail limits the number of log updates in the container file to 10 within a 3-minute period to ensure collection performance. You can change the values of the startup parameters docker_config_update_interval and max_docker_config_update_times to change the settings. For more information, see Configure the startup parameters of Logtail
File rotation for stdout and stderr	Container stdout and stderr files are rotated by Docker or kubelet. By default, the size of stdout and stderr files that are rotated by kubelet is 10 MB, and the size of stdout and stderr files that are rotated by Docker is 100 MB. If the output rate of container stdout and stderr is greater than 10 MB/s, the stdout and stderr files are rotated at a higher speed. In this case, we recommend that you collect logs from container files or change the value of the containerLogMaxSize parameter to prevent log loss.
Logging driver for stdout and stderr	If you use Docker as a container runtime, you must add "log-driver": "json-file" to the Docker configuration file daemon.json .

Checkpoints

Item	Description
Checkpoint timeout period	In default scenarios, if a log file is not modified within 30 days, the checkpoint of the log file is automatically deleted. If <pre>preserve:false</pre> is configured in a Logtail configuration and a log file is not modified within 30 minutes, the checkpoint of the log file is deleted.
Checkpoint storage policy	Checkpoints are automatically stored at intervals of 15 minutes and at the point in time when Logtail exits. You can change the value of the startup parameter check_point_dump_interval to change the checkpoint storage policy. For more information, see Configure the startup parameters of Logtail
Checkpoint storage path	By default, checkpoints are stored in the <u>/tmp/logtail_checkpoint</u> directory. You can change the value of the startup parameter check_point_filename to change the checkpoint storage path. For more information, seeConfigure the startup parameters of Logtail.

Handling during downtime	Checkpoints are saved at regular intervals. If downtime occurs,data collection resumes from the last completely saved checkpoint. This may cause duplicate data collection. You can change the checkpoint storage policy to prevent duplicate data collection.
--------------------------	--

Logtail configurations

Item	Description
Latency for configuration updates to take effect	Updates to Logtail configurations that are performed by using the Simple Log Service console or by calling an API operation require approximately 30 seconds to take effect.
Dynamic loading of Logtail configurations	Logtail configurations can be dynamically loaded. An update to a Logtail configuration does not affect other Logtail configurations.
Number of Logtail configurations that can be dynamically loaded for a single Logtail instance	Unlimited. However, we recommend that you load no more than 100 Logtail configurations for a server.
Log generation by using a third-party flusher	If a Logtail configuration is created in the Simple Log Service console or by calling an API operation, the Logtail configuration is associated with a Logstore. Therefore, when you configure a third-party flusher in your plug-in configuration, Logtail automatically sends a copy of data to the Logstore.

Machine groups

Item	Description
Number of machines	Unlimited. However, we recommend that you configure no more than 100,000 machines. Otherwise, heartbeats cannot be obtained.
Number of Logtail configurations that can be applied	Unlimited. However, we recommend that you apply no more than 1,000 Logtail configurations.

Performance

Item	Description
Throughput for log processing	The default transmission speed of raw logs is limited to 20 MB/s. Log data is uploaded after it is encoded and compressed. The compression ratio ranges from 5:1 to 10:1. If the transmission speed is faster than the default value, logs may be lost. You can change the value of the startup parameter max_bytes_per_sec to change the transmission speed. For more information, see Configure the startup parameters of Logtail
Maximum processing speed for logs	 Single-core-enabled processing speed: In simple mode, the maximum processing speed is 100 MB/s. In full regex mode, the maximum processing speed is 20 MB/s. This is the default value. The actual processing speed varies based on the complexity of regular expressions. In delimiter mode, the maximum processing speed is 40 MB/s In JSON mode, the maximum processing speed is 30 MB/s. You can configure the startup parameter process_thread_count to configure multiple threads. This helps improve performance by 150% to 300%.

	The maximum numbers of monitored directories and files are related to themem_usage_limit parameter. The default value of the mem_usage_limit parameter is 384 MB in a host environment and 2,048 MB in a container environment. The following categories are used:
	 Maximum number of monitored directories = (Value ofmem_usage_limit/100) × 5,000. The directories do not include the blacklist of directories specified in Logtail configurations.
	 Maximum number of monitored directories and files = (Value ofmem_usage_limit/100) × 50,000. The directories do not include the blacklist of directories specified in Logtail configurations. The files include the files that do not match the Logtail configurations.
	 Number of directories and files monitored by a single Logtail configuration = (Value ofmem_usage_limit/100) × 5,000. The directories do not include the blacklist of directories specified in the Logtail configuration. The files include the files that do not match the Logtail configuration.
Maximum numbers of monitored directories and files	 Number of monitored subdirectories and files in a single directory = (Value ofmem_usage_limit/100) × 5,000. The subdirectories include the blacklist of directories specified in Logtail configurations. The files include the files that do not match the Logtail configurations.
	If the number for one of the preceding categories reaches the upper limit, Logtail no longer monitors the rest of the directories and files that correspond to the category. You can narrow the scope of monitored directories in a Logtail configuration or change the value of the startup parameter mem_usage_limit to increase the number of monitored directories. For more information about the mem_usage_limit parameter, seeConfigure the startup parameters of Logtail.
	Logtail installed on a Linux server allows you to use the inotify mechanism to monitor directories. This helps shorten the latency of log collection. The startup parameter default_max_inotify_watch_num specifies the maximum number of directories (including subdirectories) that can be monitored by using the inotify mechanism. Default value: 3000.
	When the number of directories monitored by using the inotify mechanism reaches the upper limit, the regular mechanism is used to monitor the remaining directories. This does not affect log collection. You can narrow the scope of the monitored directories in a Logtail configuration or change the value of the startup parameter default_max_inotify_watch_num to increase the number of directories that can be monitored by using the inotify mechanism. For more information about the default_max_inotify_watch_num parameter, see Configure the startup parameters of Logtail.
Policy used to process excessive resource consumption	If the amount of resources occupied by Logtail remains higher than the upper limit for more than 5 minutes, Logtail is forcefully restarted. The restart may cause data loss or duplication.
Multi-tenant isolation	Logtail configurations are isolated. If an error occurs in a Logtail configuration, other Logtail configurations are not affected.
Log collection latency	In normal cases, Logtail can collect a log less than 1 second after the log is written to disk.
Log upload policy	Before Logtail uploads logs, Logtail aggregates the logs in the same file. Logtail starts to upload logs when the number of logs exceeds 2,000, the total size of logs exceeds 2 MB, or the log collection duration exceeds 3 seconds.

Error handling

Item	Description
Network error handling	 If a network error occurs, Logtail automatically retries the data collection task and adjusts the retry interval. In extreme cases, logs may be repeatedly collected or discarded due to the following issues: A packet sent by Logtail is received by Simple Log Service, and the packet responded by Simple Log Service fails to be received by Logtail within 15 seconds. In this case, Logtail sends the request again, and duplicate data is collected. A network link error causes damage to a packet received by Logtail, and the error occurs five consecutive times. In this case, Logtail discards related data.
Processing of threshold-crossing events	If a data transmission speed exceeds the upper limit of a Logstore, Logtail blocks log collection and automatically retries the data collection task. We recommend that you increase the number of shards for the Logstore.
Time errors of Logtail	If retries fail five times because the time difference between the request time and the response time is larger than 15 minutes, Logtail discards related data. A maximum of five retries are allowed. We recommend that you correct the time of your machine where Logtail resides.
Non-existence of a specified project or Logstore	If retries fail five times, Logtail discards related data. A maximum of five retries are allowed. The failure may occur if you deleted your Logstore by calling an API operation. We recommend that you delete the Logtail configurations and disassociate the Logtail configurations from your machine groups by calling API operations.
Failed authentication	 If retries fail five times, Logtail discards related data. A maximum of five retries are allowed. The failure may occur in the following scenarios: If the failure occurs when Logtail is started, the cause is that no authentication information can be obtained over an unstable network. If the failure repeatedly occurs, the cause is that no authentication information can be obtained when your machine cannot connect to a Simple Log Service endpoint over HTTPS.
Other unknown errors	If retries fail five times, Logtail discards related data. A maximum of five retries are allowed.
Maximum retry period before timeout	If data fails to be transmitted and the issue lasts for more than 6 hours, Logtail discards the data.
Status self-check	If an exception occurs, Logtail restarts. For example, if an application unexpectedly exits or the resource usage exceeds the specified upper limit, Logtail restarts.

4.2.9.4. Data import

4.2.9.4.1. Limits on data import from OSS to Simple Log Service

This topic describes the limits on data import from Object Storage Service (OSS) to Simple Log Service.

Limits on collection

Item	Description
Size of a single object	 A data import job can import logs from an object that is Snappy-compressed without using a framing format. The maximum size of the object can be 350 MB. A data import job can import logs from an object that is in other formats. The maximum size of the object can be 5 GB. If the size of a single object exceeds the limit, the entire object is ignored during import.
Size of a single data record	The maximum size of a single data record can be 3 MB. If the size of a single data record exceeds the limit, the record is discarded. You can view the number of data records that are discarded in the Deliver Failed chart on the Data Processing Insight dashboard.
Object update	Data import jobs import full data of updated OSS objects to Simple Log Service. If new data is appended to an OSS object that is imported to Simple Log Service, all data of the OSS object is re-imported to Simple Log Service when a data import job for the OSS object is run.
Detection latency of new objects	The minimum interval at which a data import job detects new objects is 1 minute. If a data import job writes a large number of objects to Simple Log Service, high latency may exist.

Limits on configuration

Item	Description
Number of data import configurations	A maximum of 100 data import configurations can be created in a single project regardless of configuration types. If you want to increase the quota, contact Alibaba Cloud technical support.

Limits on performance

Item	Description
Number of concurrent subjobs	Simple Log Service automatically creates multiple data import subjobs to concurrently import data based on the number of objects that need to be imported. Simple Log Service automatically creates a maximum of eight subjobs for each data import configuration. Each subjob can process decompressed data at a maximum speed of 10 MB/s. In total, a data import job can process decompressed data at a maximum speed of 80 MB/s. If you want to increase the quota, contact Alibaba Cloud technical support.
Number of shards in a Logstore	The write performance of Simple Log Service varies based on the number of shards in a Logstore. A single shard supports a maximum write speed of 5 MB/s. If a data import job writes a large amount of data to Simple Log Service, we recommend that you increase the number of shards in the Logstore. For more information, see Manage shards.
Data read from Archive objects	If the objects that you want to import are Archive objects, you must restore the objects before Simple Log Service can read data from the objects. In most cases, Archive objects require approximately 1 minute to be restored.
Size of objects	If the total volume of data is the same, the larger the size of each object, the higher the read throughput. The smaller the size of each object, the lower the read throughput.
Network	If your OSS bucket and Simple Log Service project reside in the same region, no Internet traffic is generated, and data is transferred at a high speed. If your OSS bucket and Simple Log Service project reside in different regions, the object read is significantly affected by network conditions, and the read performance is relatively poor.
Import latency of new data	If the number of existing objects is large and you do not enable OSS Metadata Indexing , the value of the New File Check Cycle parameter may not take effect when new objects are imported to Simple Log Service. If the number of existing objects is approximately 1 million, the import latency of new data is approximately 2 minutes. The import latency varies in a linear manner with the number of existing objects.

4.2.9.5. Data transformation

This topic describes the limits on data transformation in Simple Log Service.

Job configuration

Item	Description
Job quantity	You can create a maximum of 100 data transformation jobs in a project. Important When a data transformation job is stopped or complete, the job still consumes the job quota. To prevent the quota from being consumed by the data transformation jobs that are stopped or complete, we recommend that you delete the jobs that you no longer use. For more information, see Manage a data transformation job. If you want to increase the quota, contact Alibaba Cloud technical support.
Dependency of a consumer group in a source Logstore	The running of a data transformation job depends on a consumer group in the source Logstore. When a data transformation job is running, do notdelete or reset the consumption checkpoint for the consumer group on which the job depends. If you perform the delete or reset operation, the job consumes data again from the start time that you specify, and duplicate data exists in the result. Important The data consumption progress of a job in a shard is updated to the consumer group on which the job depends at regular intervals. This optimizes the efficiency of data transformation. However, the result of the GetCheckPoint operation on the consumer group cannot indicate the latest data transformation progress. To obtain the accurate data transformation progress of a job, you can go to the shard consumption delay chart of the dashboard that is created for the job. For more information, see Data transformation dashboard. For more information, see Data transformation basics, Terms, and Consumer group operations.
Number of consumer groups in a source Logstore	You can create a maximum of 30 consumer groups in a Logstore. This way, you can create a maximum of 30 data transformation jobs in a source Logstore. For more information, see Basic resources. Important When a data transformation job is stopped or complete, Simple Log Service does not automatically delete the consumer group on which the job depends. To reduce invalid consumer groups, we recommend that you delete the data transformation jobs that are stopped or complete and you no longer use. For more information, see Manage a data transformation job.
Modification of time ranges for jobs	 If you modify the time range for a running job, the job starts consumption from the start time that you specify and consumes all data that is generated in the newly specified time range. 1. If you want a job to consume data that is generated within a longer time range, we recommend that you create another job to expand the time range instead of prolonging the time range of the existing job. 2. If you want a job to consume data that is generated within a shorter time range, we recommend that you delete the data that is written to the storage destinations and then shorten the time range of the existing job to prevent data duplication. The data that is written to the storage destinations is not automatically deleted.
Number of storage destinations	You can configure a maximum of 20 independent static storage destinations for a data transformation job. A maximum of 200 projects and 200 Logstores can be dynamically specified in data transformation code. If one of the preceding limits is exceeded, the data that is written to a different storage destination other than the allowed 20 storage destinations is discarded.

Data transformation

Item	Description
Quick preview	 The quick preview feature of data transformation is used to debug data transformation code. The feature has the following limits: Connections to external resources such as ApsaraDB RDS, Object Storage Service (OSS), and Simple Log Service are not supported. You can specify custom test data for a dimension table. A single request can obtain no more than 1 MB of test data from a source table or a dimension table. If the size of the data exceeds 1 MB, an error is returned. A maximum of the first 100 logs can be returned for a single request. The advanced preview feature does not have the limits.
Runtime concurrency	 The number of readwrite shards in a source Logstore specifies the maximum number of data transformation jobs that can concurrently run. For more information, see Data transformation basics. For more information about the limits on shards, seeBasic resources. For information about how to split a shard for a Logstore, see Manage shards. Important If the number of data transformation jobs that can concurrently run does not meet the requirements, automatic sharding is not triggered for the source Logstore, and you must manually split a shard of the source Logstore to increase the number of data transformation jobs that can concurrently run. For more information about automatic sharding, see Manage shards. For data that is written after the shard is split, the maximum number of data transformation jobs that can concurrently run equals the number of readwrite shards that are available in the source Logstore after splitting. For data that is written before the shard is split, the maximum number of data transformation jobs that can concurrently run equals the number of readwrite shards that are available in the source Logstore when the data is written.

Data load of a concurrent unit	The data load of a concurrent unit in a data transformation job is determined by the amount of data that is consumed by the job from a shard of the source Logstore. If the data that is written to the source Logstore is unevenly distributed among the shards of the Logstore, hot concurrent units may occur when the data transformation job is running. This may cause processing delays on some shards. If data is written to the source Logstore in KeyHash mode, we recommend that you appropriately allocate hash keys and shards to minimize unbalanced data distribution. For more information about data write, see <i>PutLogs</i> in <i>Simple Log Service Developer Guide > API Reference</i> .	
Memory usage	The memory usage threshold for a concurrent unit in a data transformation job is 6 GB. If the memory occupied by a concurrent unit exceeds 6 GB, the performance of the job is limited, and processing latency exists. The memory occupied by a concurrent unit exceeds the threshold when a large number of log groups are pulled at a time. You can modify the advanced parameter system.process.batch_size to adjust the memory usage threshold. () Important The maximum value allowed for the advanced parameter system.process.batch_size is 1000. You can change the value to a positive integer that is less than or equal to 1,000. The default value of system.process.batch_size is 1000.	
CPU utilization	The CPU utilization threshold for a concurrent unit of a data transformation job is 100%. If you have higher requirements for CPU utilization, you can increase the number of data transformation jobs that can concurrently run based on the preceding descriptions.	
Data amount in a dimension table	The maximum number of data entries allowed in a dimension table is 2 million, and the maximum memory that can be occupied by data in a dimension table is 2 GB. If the number of data entries exceeds 2 million or the memory occupied by data exceeds 2 GB, truncation is performed. In this scenario, only the 2 million data entries in a dimension table or the data that occupies the 2 GB of memory for a dimension table can be used. The functions that are involved include res_rds_mysql, res_log_logstore_pull, and res_oss_file.	
	Important If a single data transformation job consumes data from multiple dimension tables, the tables must conform to the limits as a whole. We recommend that you minimize the amount of data in a dimension table.	

Result data writing

Item	Description
Data writing to a destination Logstore	When transformation results are written to a destination Logstore, the write limits of the Logstore cannot be exceeded. For more information, see Basic resources and Data read and write. If you configure the hash_key_field or hash_key parameter and specify the KeyHash mode when you call the e_output and e_coutput functions to write data to a destination Logstore, we recommend that you appropriately allocate hash keys and shards to minimize unbalanced data transformation job are written to a destination Logstore, repeated retries are performed to ensure that the transformation log is compromised, and the processing of data in the source shard is delayed.
Cross-region data transmission	When data is transferred across regions by using a public endpoint, network quality cannot be ensured. In this case, a network error may occur when the results of a data transformation job are written to a destination Logstore. This delays the progress of the entire data transformation job. For more information about Simple Log Service endpoints, see <i>Obtain the endpoint of Simple Log</i> <i>Service</i> in <i>Simple Log Service Developer Guide > Preparations</i>

4.2.9.6. Query and analysis

This topic describes the limits on query and analysis in Simple Log Service.

Query

Item	Description	Remarks
Number of keywords	The number of keywords that are used as search conditions. The number of logical operators is not included. You can specify a maximum of 30 keywords in a search statement.	None
Size of a field value	The maximum size of a field value is 10 KB. The excess part is not involved in searching.	If the size of a field value is greater than 10 KB, logs may fail to be obtained by using keywords, but the logs are actually stored in the Logstore.
Number of concurrent search statements	Each project supports a maximum of 100 concurrent search statements.	For example, 100 users can concurrently execute search statements in all Logstores of a project.

Cloud Defined Storage

Returned result	The returned logs are displayed on multiple pages. Each page displays a maximum of 100 logs.	None
Maximum size of a log	Simple Log Service performs the Document Object Model (DOM) operation only on the first 10,000 characters of a log due to browser performance limits.	If a log contains more than 10,000 characters, the following message appears in the Simple Log Service console: The log contains log data of more than 10,000 characters, and the display of some characters will be downgraded.
Fuzzy search	In a fuzzy search, Simple Log Service matches up to 100 words that meet the specified conditions and returns the logs that meet the search conditions and contain one or more of these words. For more information, see Search syntax.	None
Data sorting in search results	By default, search results are displayed in descending order of time, which is accurate to minutes.	None

Analysis

Item	Standard SQL	Dedicated SQL
Number of concurrent analytic statements	Each project supports a maximum of 15 concurrent analytic statements. For example, 15 users can concurrently execute analytic statements in all Logstores of a project.	Each project supports a maximum of 100 concurrent analytic statements. For example, 100 users can concurrently execute analytic statements in all Logstores of a project.
Data volume	Each shard supports only 1 GB of data for a single analytic statement.	An analytic statement can scan a maximum of 200 billion rows of data.
Method to enable	By default, Standard SQL is enabled.	A switch is provided for you to manually enable Dedicated SQL. For more information, see Enable Dedicated SQL.
Resource usage fee	Free of charge.	You are charged based on the actual CPU time.
Applicable scope	You can analyze only the data that is written to Simple Log Service after the log analysis feature is enabled. If you want to analyze historical data, you must reindex the historical data. For more information, see Reindex logs for a Logstore.	You can analyze only the data that is written to Simple Log Service after the log analysis feature is enabled. If you want to analyze historical data, you must reindex the historical data. For more information, see Reindex logs for a Logstore.
Returned result	By default, an analytic statement returns a maximum of 100 rows of data. If you want to view more data, use a LIMIT clause. For more information, see LIMIT clause.	By default, an analytic statement returns a maximum of 100 rows of data. If you want to view more data, use a LIMIT clause. For more information, see LIMIT clause.
Size of a field value	The maximum length of a field value that can be retained for analysis is 16,384 bytes, which is equivalent to 16 KB. O Note By default, the maximum length of a field value that can be retained for analysis is 2,048 bytes, which is equivalent to 2 KB. If you want to change the maximum length of a field value, you can configure the Maximum Statistics Field Length parameter. For more information, see Configure indexes.	The maximum length of a field value that can be retained for analysis is 16,384 bytes, which is equivalent to 16 KB. ⑦ Note By default, the maximum length of a field value that can be retained for analysis is 2,048 bytes, which is equivalent to 2 KB. If you want to change the maximum length of a field value, you can configure the Maximum Statistics Field Length parameter. For more information, see Configure indexes.
Timeout period	The maximum timeout period for a single analytic statement is 55 seconds.	The maximum timeout period for a single analytic statement is 55 seconds.
Number of bits in the mantissa part of a double-type field value	A double-type field value can contain a maximum of 52 bits in the mantissa part. If the mantissa part of a double-type field value contains more than 52 bits, the precision of the field value is compromised.	A double-type field value can contain a maximum of 52 bits in the mantissa part. If the mantissa part of a double-type field value contains more than 52 bits, the precision of the field value is compromised.

4.2.9.7. Scheduled SQL

This topic describes the limits of the Scheduled SQL feature.

Special jobs

Some applications such as Trace of Simple Log Service depend on the Scheduled SQL feature. To ensure that these applications can be used as expected, Scheduled SQL does not allow any changes to the jobs that are generated when you use the applications. These jobs are called special jobs. You cannot update, copy, or delete a special job on the Scheduled SQL page. If you want to update, copy, or delete a special job, perform the operation in the related application.

Query and analysis

! Important

Scheduled SQL supports only Dedicated SQL.

Item	Description
Number of concurrent analytic statements	Each project supports a maximum of 150 concurrent analytic statements. For example, 150 users can concurrently execute analytic statements in all Logstores of a project.
Data volume	An analytic statement can scan a maximum of 200 billion rows of data.
Applicable scope	You can analyze only the data that is written to Simple Log Service after the log analysis feature is enabled. If you want to analyze historical data, you must reindex the historical data. For more information, see Reindex logs for a Logstore.
Return result	 By default, an analytic statement returns a maximum of 100 rows of data. Excess data is not returned. If you want an analytic statement to return more data, you can use the LIMIT clause in the statement. A maximum of 1 million rows of data can be returned. For more information, see LIMIT clause. Data beyond the range specified by the LIMIT clause is not returned. The volume of data that an analytic statement can return is limited to 20 GB. Excess data is not returned.
Size of a field value	By default, the size of a field value is 2,048 bytes, equivalent to 2 KB. The maximum size of a field value is 16,384 bytes, equivalent to 16 KB. If the size of a field value exceeds 16 KB, the excess data is not involved in analysis. You can modify the maximum size for a field value when you configure indexes. Valid values: 64 to 16384. Unit: bytes. For more information, see Configure indexes.
Timeout period	The maximum timeout period for an analytic statement is 10 minutes.
Number of bits in the mantissa part of a double-type field value	A double-type field value can contain a maximum of 52 bits in the mantissa part. If the mantissa part of a double-type field value contains more than 52 bits, the precision of the field value is compromised.
Fuzzy search	In a fuzzy search, Simple Log Service matches a maximum of 100 words that meet the specified conditions and returns the logs that contain one or more of these words and meet the query conditions.
Inaccurate query result	If query results are inaccurate, no errors are reported. However, the issue is recorded in the instance status information and included in job running records. The recording feature must be manually enabled.
Data latency	If data latency occurs, some data may be missed in query. If the data of a point in time arrives later after the instance for that point in time runs, the data is not included when the next instance runs.
Time window	The time window for a single query ranges from 1 minute to 24 hours.
Metastore association	Not supported.
LIMIT clause	Scheduled SQL supports only $\mbox{ LIMIT x }$. Scheduled SQL does not support $\mbox{ LIMIT y, x }$.

Data write

Item	Description
Write threshold of a Logstore	If the write threshold is exceeded when you write data, the Scheduled SQL job is retried for more than 10 minutes. After the retry time, an error message is returned. For more information, see Data read and write.
Cross-region data transmission	When data is transmitted across regions inside China, the network is stable, but latency is high. The latency varies based on regions. When data is transmitted across regions outside China, network quality cannot be ensured.

Job running

Item	Limit
Timeout period	The maximum timeout period of a job is 1,800 seconds. If the timeout period of a job is exceeded, the job is considered failed. We recommend that you create an alert monitoring task to detect errors and retry failed instances in a timely manner. For more information, see Retry an instance of the Scheduled SQL job

Number of retries	The maximum number of retries for a job is 100. If a job is retried for more than 100 times, the job is considered failed.
Delayed running	You can delay running an instance for a maximum of 120 seconds. For more information about delayed running scenarios, see Scheduling and running scenarios.
Historical running record	The historical running records of a single job can be stored for a maximum of 14 days. We recommend that you create an alert monitoring task to detect errors and retry failed instances in a timely manner. For more information, see Retry an instance of the Scheduled SQL job

4.2.9.8. Alerting

This topic describes the limits of the alerting feature in Simple Log Service.

Alerting

Item	Description	
Associated query statements	You can associate an alert rule with a maximum of three query statements.	
String	If a field value exceeds 1,024 characters in length, Simple Log Service extracts only the first 1,024 characters for data processing.	
Conditional expressions	 The trigger condition has the following limits: Each trigger condition must be 1 to 128 characters in length. If a query result includes more than 100 rows, Simple Log Service only checks whether the first 100 rows meet the trigger condition. Simple Log Service checks whether a trigger condition is met for a maximum of 1,000 times for the specified query statements. 	
Query time range	The maximum time range that you can specify for each query is 24 hours.	
Voice calls	If a voice call is not answered, Simple Log Service sends a text message. You are charged for a voice call regardless of whether the call is answered. You are not charged for the text message that is sent upon a non-answered voice call.	

4.3. Data collection

4.3.1. Collection by Logtail

4.3.1.1. Overview

4.3.1.1.1. Logtail overview

Logtail is a log collection agent that is provided by Log Service. You can use Logtail to collect logs from multiple data sources in real time. These sources include Elastic Compute Service (ECS) instances, data centers, and servers that belong to third-party cloud service providers. This topic describes the features, benefits, limits, and configuration process of Logtail.

Figure 1. Logtail-based log collection



Benefits

- Supports non-intrusive log collection based on log files. You do not need to modify your application code. Your applications are not affected when Logtail collects logs.
- Allows you to collect text logs, binary logs, HTTP data, and container logs.
- Allows you to collect logs from standard containers, swarm clusters, and Kubernetes clusters.
- Handles exceptions during log collection. If a network or server exception occurs, Logtail retries log collection and caches logs on local servers to ensure data security.
- Provides centralized management based on Log Service. After you install Logtail on servers and create a machine group and Logtail configurations, Logtail collects logs from the servers and sends the logs to Log Service.
- Provides a comprehensive self-protection mechanism. The CPU, memory, and network resources that Logtail can use are limited. This ensures that Logtail does not affect the performance of other services on the server.

Limits

For more information about the limits of Logtail, see Limits.

Configuration process



To collect logs from servers by using Logtail, perform the following steps:

1. Install Logtail.

Install Logtail on servers from which you want to collect logs. For more information, see Install Logtail on a Linux server and Install Logtail on a Windows server.

2. Create a machine group.

Log Service allows you to create a custom ID-based machine group or an IP address-based machine group. For more information, see Create an IP address-based machine group and Create a custom identifier-based machine group.

3. Create a Logtail configuration and apply it to the machine group.

After you complete the preceding procedure, Logtail collects logs from your server and sends the logs to the specified Logstore. You can use the Log Service console, call API operations, or use SDKs to query logs.

Terms

• Machine group: A machine group contains one or more servers from which logs of a specific type are collected. After you apply Logtail configurations to a machine group, Log Service collects logs from the servers in the machine group based on the configurations.

You can set an IP address-based identifier or a custom identifier for a machine group. Then, you can manage the servers in the machine group based on the identifier. You can create and delete a machine group, add servers to a machine group, and remove servers from a machine group in the Log Service console.

- Logtail: Logtail is a log collection agent that is provided by Log Service. Logtail runs on servers to collect logs from the servers. For more information, see Install Logtail on a Linux server or Install Logtail on a Windows server.
 - For a Linux-based server, Logtail is installed in the /usr/local/ilogtail directory. Logtail initiates the following processes whose names start with
 ilogtail: a log collection process and a daemon process. The logs of Logtail are stored in the /usr/local/ilogtail/ilogtail.LOG file.
- For a Windows-based server, Logtail is installed in the C:\Program Files\Alibaba\Logtail directory (32-bit system) or C:\Program Files
 (x86)\Alibaba\Logtail directory (64-bit system). Choose Control Panel > Administrative Tools > Services. On the Services window, you can
 view the LogtailDaemon service. The logs of Logtail are stored in the ilogtail.LOG file.
- Logtail configurations: Logtail configurations are a set of policies that Logtail uses to collect logs. You can specify the data source and collection
 mode to create custom Logtail configurations for log collection. The configurations specify how to collect logs from servers, parse the logs, and send
 the logs to a specified Logstore.

Features

Feature	Description	
Real-time log collection	Logtail monitors log files, and reads and parses incremental logs in real time. In most cases, logs are sent to Log Service within 3 seconds after they are generated. Note Logtail does not collect historical data. If Logtail reads a log later than 12 hours after the log was generated, Logtail drops the log.	
Automatic log rotation	Multiple applications rotate log files based on the file size or date. The original log file is renamed and an empty log file is created during the rotation process. For example, the app.LOG file is renamed app.LOG.1 and app.LOG.2 during log rotation. You can specify the file to which collected logs are written, for example, app.LOG. Logtail monitors the log rotation process to ensure that no logs are lost.	
Multiple data sources	Logtail can collect text logs, syslogs, HTTP logs, and MySQL binlogs.	

Compatibility with open source collection agents	You can use open source agents such as Logstash and Beats to collect data. Then, you can use Logtail to collect data from the agents and send the data to Log Service.	
Automatic exception handling	If data fails to be sent to Log Service due to exceptions, Logtail retries to collect logs based on the scenario. The exceptions include server errors, network errors, and quota exhaustion. If the retry fails, Logtail writes the data to the local cache and resends the data after 3 seconds.	
Flexible collection policy configuration	Logtail allows you to create configurations for log collection in a flexible manner. You can specify the directories and files from which logs are collected. You can also specify an exact match or a wildcard match based on your business requirements. You can also specify the log collection mode and customize the fields that you want to extract. You can use a regular expression to extract fields from logs. Log data in Log Service must have the timestamp information. Logtail allows you to customize log time formats and then extract the required timestamps from the time information based on different formats.	
Automatic synchronization of Logtail configurations	After you create or update Logtail configurations in the Log Service console, the configurations are synchronized to the servers in which Logtail is installed and take effect within 3 minutes. Logs are collected based on the original configurations during the synchronization.	
Status monitoring	Logtail monitors the CPU and memory resources that are consumed in real time. This ensures that Logtail does not consume an excessive number of resources or affect other services. If the resource consumption exceeds the limit, Logtail is automatically restarted. Logtail also monitors the network bandwidth resources that are consumed. This ensures that Logtail does not consume an excessive amount of bandwidth.	
Data transmission with a signature	Logtail retrieves the AccessKey pair of your Apsara Stack tenant account and uses the pair to sign all log data that is sent to Log Service. This way, data tampering is prevented during data transmission. ⑦ Note Logtail obtains the AccessKey pair of your Apsara Stack tenant account by using the HTTPS protocol to ensure the security of your AccessKey pair.	

Data collection reliability

Logtail stores checkpoints that are periodically collected to the local server during log collection. If an exception such as an unexpected server shutdown or a process failure occurs, Logtail restarts and then collects data from the last checkpoint. This process avoids incomplete data collection. Logtail runs based on the startup parameters that are specified in the startup configuration file. If the usage of a resource exceeds the limit for more than 5 minutes, Logtail is forcibly restarted. After the restart, a small amount of duplicate data may be collected to the specified Logstore.

To improve log collection reliability, Logtail uses multiple internal mechanisms. However, logs may fail to be collected in the following scenarios:

- Logtail is not running, but logs are rotated multiple times.
- The log rotation rate is high, for example, one rotation per second.
- The log collection rate is lower than the log generation rate for a long period of time.

4.3.1.1.2. Log collection process of Logtail

This topic describes how Logtail collects logs. The log collection process consists of the following steps: monitor log files, read log files, process logs, filter logs, aggregate logs, and send logs.

Monitor log files

After you install Logtail on a server and create a Logtail configuration that is used to collect logs in the Log Service console, Log Service delivers the Logtail configuration to Logtail in real time. Then, Logtail monitors log files of the server based on the Logtail configuration. Logtail scans log directories and files based on the log file path and the maximum directory depth that you specify for monitoring in the Logtail configuration.

If the log files of the server in a machine group are not updated after you apply the Logtail configuration to the machine group, the log files are considered historical log files. Logtail does not collect historical log files. If log files are updated, Logtail reads and collects the files, and then sends the log files to Log Service. For more information about how to collect historical log files, see Import historical log files.

Logtail registers event listeners to monitor directories from which log files are collected. The event listeners pool the log files in the directories on a regular basis. This ensures that logs are collected at the earliest opportunity in a stable manner. For Linux-based servers, **Inotify** is used to monitor the directories and pool log files.

Read log files

After Logtail detects updated log files, Logtail reads the log files.

- The first time Logtail reads a log file, Logtail checks the size of the file.
 - If the file size is less than 1 MB, Logtail reads data from the beginning of the file.
 - If the file size is greater than 1 MB, Logtail reads from the last 1 MB of data in the file.
- If a log file is read before, Logtail reads the file from the previous checkpoint.
- Logtail can read up to 512 KB of data at the same time. Make sure that the size of each log in a log file does not exceed 512 KB.

() Important

If you change the system time on the server, you must restart Logtail. Otherwise, the log time becomes invalid and logs are dropped.

Process logs

After Logtail reads a log file, Logtail splits each log in the file into multiple lines, parses the log, and then configures the time field for the log.

• Split a log into multiple lines

If you specify a regular expression to match the start part in the first line of a log, Logtail splits the log into multiple lines based on the regular expression. If you do not specify a regular expression, a single log line is processed as a log.

Parse logs

Logtail parses each log based on the collection mode that you specify in the Logtail configuration.

? Note

If you specify complex regular expressions, Logtail may consume an excessive amount of CPU resources. We recommend that you specify regular expressions that allow Logtail to parse logs in an efficient manner.

- If Logtail fails to parse a log, Logtail handles the failure based on the setting of the Drop Failed to Parse Logs parameter in the Logtail configuration.
- $\circ~$ If you turn on $\mbox{Drop Failed to Parse Logs}$, Logtail drops the log and reports an error.
- If you turn off Drop Failed to Parse Logs, Logtail uploads the log. The key of the log is set to raw_log and the value is set to the log content.
 Configure the time field for a log
 - If you do not configure the time field for a log, the log time is set to the time when the log is parsed.
 - If you configure the time field for a log, Logtail processes the log based on the following conditions:
 - If the difference between the time when the log is generated and the current time is within 12 hours, the log time is extracted from the parsed log fields.
 - If the difference between the time when the log is generated and the current time is greater than 12 hours, the log is dropped and an error is reported.

Filter logs

After logs are processed, Logtail filters the logs based on the specified filter conditions.

- If you do not specify filter conditions in the Filter Configuration field, the logs are not filtered.
- If you specify filter conditions in the **Filter Configuration** field, the fields in each log are traversed. Logtail collects only the logs that meet the filter conditions.

Aggregate logs

To reduce the number of network requests, Logtail caches the processed and filtered logs for a specified period of time. Then, Logtail aggregates the logs and sends the logs to Log Service.

If one of the following conditions is met during the cache process, logs are aggregated and sent to Log Service:

- The aggregation duration exceeds 3 seconds.
- The number of aggregated logs exceeds 4,096.
- The total size of aggregated logs exceeds 512 KB.

Send logs

Logtail sends the aggregated logs to Log Service. You can set the **max_bytes_per_sec** and **send_request_concurrency** parameters in the Logtail startup configuration file to specify the maximum transmission rate of log data and concurrent requests. For more information, see Configure the startup parameters of Logtail.

If a log fails to be sent, Logtail retries or no longer sends the log based on the error code.

Error code	Description	Handling method
401	Logtail is not authorized to collect data.	Logtail drops the log packets.
404	The project or Logstore that is specified in the Logtail configuration does not exist.	Logtail drops the log packets.
403	The shard quota is exhausted.	Logtail tries again after 3 seconds.
500	A server exception occurs.	Logtail tries again after 3 seconds.

4.3.1.1.3. Logtail configuration files and record files

This topic describes the basic configuration files and record files of Logtail. When Logtail is active, Logtail uses the configuration files and generates record files.

Startup configuration file (ilogtail_config.json)

The ilogtail_config.json file is used to set the startup parameters of Logtail. For more information, see Configure the startup parameters of Logtail.

⑦ Note

- The file must be a valid JSON file. Otherwise, Logtail cannot be started.
- If you modify the file, you must restart Logtail for the modifications to take effect.

After you install Logtail on a server, you can perform the following operations on the ilogtail_config.json file:

- Modify the runtime parameters of Logtail.
- Check whether the installation commands are correct.
- The values of the **config_server_address** and **data_server_list** parameters in the ilogtail_config.json file vary based on the installation command that you select. If the region in the command is different from the region where the Log Service project resides or the address in the command cannot be accessed, the command is incorrect. If the command is incorrect, Logtail cannot collect logs and must be reinstalled.
- File path
- Linux: The file is stored in /usr/local/ilogtail/ilogtail_config.json.
• Windows:

- 64-bit: The file is stored in C:\Program Files (x86)\Alibaba\Logtail\ilogtail_config.json.
- 32-bit: The file is stored in C:\Program Files\Alibaba\Logtail\ilogtail_config.json.

? Note

You can run both 32-bit and 64-bit applications in a 64-bit Windows operating system. To ensure compatibility, the operating system stores 32bit applications in a separate x86 directory.

Logtail for Windows is a 32-bit application. Therefore, Logtail is installed in the Program Files (x86) directory in 64-bit Windows. If Logtail for 64-bit Windows is available, you can install Logtail in the Program Files directory.

• Containers: The file is stored in a Logtail container. The file path is specified in the environment variable ALIYUN_LOGTAIL_USER_ID of the Logtail container. You can run the docker inspect {logtail_container_name} | grep ALIYUN_LOGTAIL_CONFIG command to view the file path. Example: /etc/ilogtail/conf/cn-hangzhou/ilogtail_config.json.

Sample file

If you run the cat /usr/local/ilogtail_ingtail_config.json command, the following output is returned:

User identifier file

The user identifier file contains the ID of your Apsara Stack tenant account. The file specifies that the account is authorized to collect logs from the server on which Logtail is installed. For more information, see Configure a user identifier.

? Note

- If the server is an Elastic Compute Service (ECS) instance that belongs to another Apsara Stack tenant account, a server that is deployed in a self-managed data center, or a server that is provided by a third-party cloud service provider, you must specify the ID of your Apsara Stack tenant account as a user identifier for the server after you install Logtail. Then, you can use Logtail to collect logs from the server by using the account.
- You must specify the ID of an Apsara Stack tenant account as a user identifier in the user identifier file. You cannot specify the ID of a RAM user as a user identifier.
- You must specify the name of the user identity file. You do not need to specify the file extension.
- You can specify multiple user identifiers on a server. However, you can specify only one user identifier for a Logtail container.
- File path
 - Linux: The file is stored in /etc/ilogtail/users/.
 - Windows: The file is stored in C:\LogtailData\users\.
 - Containers: The file is stored in a Logtail container. The file path is specified in the environment variable **ALIYUN_LOGTAIL_USER_ID** of the Logtail container. You can run the docker inspect \${logtail_container_name} | grep ALIYUN_LOGTAIL_USER_ID command to view the file path.
- Sample file

If you run the ls /etc/ilogtail/users/ command, the following output is returned:

782392********* 37292***********

Custom identifier file (user_defined_id)

The user_defined_id file is used to configure a custom ID for a machine group. For more information, see Create a custom identifier-based machine group.

? Note

When you create a custom ID-based machine group, you must configure the user_defined_id file.

- Linux: The file is stored in /etc/ilogtail/user_defined_id.
- Windows: The file is stored in C:\LogtailData\user_defined_id.
- Containers: The file is stored in a Logtail container. The file path is specified in the environment variable **ALIYUN LOGTAIL_USER_DEFINED_ID** of the Logtail container. You can run the docker inspect \${logtail_container_name} | grep ALIYUN_LOGTAIL_USER_DEFINED_ID command to view the file path.
- Sample file
 - If you run the cat /etc/ilogtail/user_defined_id command, the following output is returned:

aliyun-ecs-rs1e16355

[•] File path

Logtail configuration file (user_log_config.json)

The user_log_config.json file records the information of a Logtail configuration received by Logtail from Log Service. The file is in the JSON format and is updated along with configuration updates. You can use the user_log_config.json file to check whether the Logtail configuration is delivered to the server on which Logtail is installed. If the Logtail configuration file exists and the configurations in the file are the same as the settings of the Logtail configuration is delivered.

? Note

We recommend that you do not modify the Logtail configuration file unless you need to specify sensitive information, such as the AccessKey pair and database password.

- File path
- Linux: The file is stored in /usr/local/ilogtail/ilogtail_config.json.
- Windows
 - 64-bit: The file is stored in C:\Program Files (x86)\Alibaba\Logtail\user_log_config.json.
 - 32-bit: The file is stored in C:\Program Files\Alibaba\Logtail\user_log_config.json.
- Containers: The file is stored in /usr/local/ilogtail/user_log_config.json.

Sample file

```
"metrics" : {
   "##1.0##k8s-log-c12ba2028****939f0b$app-java" : {
      "aliuid" : "16542189*****50",
      "category" : "app-java",
"create_time" : 1534739165,
      "defaultEndpoint" : "cn-hangzhou-intranet.log.aliyuncs.com",
      "delay_alarm_bytes" : 0,
      "enable" : true,
      "enable_tag" : true,
      "filter_keys" : [],
      "filter_regs" : [],
      "group_topic" : "",
      "local_storage" : true,
      "log_type" : "plugin",
"log_tz" : "",
      "max_send_rate" : -1,
      "merge_type" : "topic",
      "plugin" : {
          "inputs" : [
            {
                "detail" : {
                   "IncludeEnv" : {
                      "aliyun_logs_app-java" : "stdout"
                   },
                   "IncludeLable" : {
                      "io.kubernetes.container.name" : "java-log-demo-2",
                      "io.kubernetes.pod.namespace" : "default"
                   },
                   "Stderr" : true,
                   "Stdout" : true
                },
                "type" : "service_docker_stdout"
            }
         ]
      },
       "priority" : 0,
      "project_name" : "k8s-log-c12ba2028c*****ac1286939f0b",
      "raw_log" : false,
      "region" : "cn-hangzhou",
      "send_rate_expire" : 0,
      "sensitive_keys" : [],
      "tz_adjust" : false,
      "version" : 1
}
```

AppInfo record file (app_info.json)

The app_info.json file records the information of Logtail, such as the startup time and the IP address and hostname obtained by Logtail.

If the IP address of a server is associated with the hostname in the /etc/hosts file of the server, Logtail obtains the IP address. If you do not associate the IP address of a server with the hostname, Logtail obtains the IP address of the first network interface controller (NIC) on the server.

? Note

}

- The AppInfo record file records only the basic information of Logtail.
- If you modify the hostname or other network settings of the server, you must restart Logtail to obtain a new IP address.

• File path

[•] Linux: The file is stored in /usr/local/ilogtail/app_info.json.

User Guide-Log Service

- Windows
 - 64-bit: The file is stored in C:\Program Files (x86)\Alibaba\Logtail\app_info.json.
- 32-bit: The file is stored in C:\Program Files\Alibaba\Logtail\app_info.json.
- Containers: The file is stored in the /usr/local/ilogtail/app_info.json.
- Sample file

If you run the cat /usr/local/ilogtail/app_info.json command, the following output is returned:

```
{
    "UUID" : "",
    "hostname" : "logtail-ds-slpn8",
    "instance_id" : "E5F93EC6-B024-11E8-8831-0A58AC14039E_1**.***.***.1536053315",
    "ip" : "1**.***.****,
    "logtail_version" : "0.16.13",
    "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
    "update_time" : "2018-09-04 09:28:36"
}
```

Field	Description
UUID	The serial number of the server.
hostname	The hostname.
instance_id	The unique identifier of Logtail. This identifier is randomly generated.
ip	The IP address that is obtained by Logtail. If Logtail does not obtain an IP address, the value of this parameter is null. Logtail cannot run as expected. You must specify an IP address for the server and then restart Logtail. O Note If you create an IP address-based machine group, you must make sure that the IP address that you specify for the machine group is the same as the value of this field. If the two IP addressed are different, modify the IP address that you specify for the machine group in the Log Service console. Check the IP addresses again after 1 minute.
logtail_version	The version of Logtail.
os	The version of the operating system.
update_time	The last startup time of Logtail.

Logtail operational log file (ilogtail.LOG)

The logtail_plugin.LOG file records the operational logs of Logtail plug-ins. The levels of logs in ascending order include INFO, WARN, ERROR. You can ignore logs at the INFO level.

- File path
- Linux: The file is stored in /usr/local/ilogtail/ilogtail.LOG.
- Windows
 - 64-bit: The file is stored in C:\Program Files (x86)\Alibaba\Logtail\ilogtail.LOG.
 - 32-bit: The file is stored in C:\Program Files\Alibaba\Logtail\ilogtail.LOG.
- Containers: The file is stored in /usr/local/ilogtail/ilogtail.LOG.
- Sample file

If you run the tail /usr/local/ilogtail.LOG command, the following output is returned:

[2018-09-13 01:13:59.024679]	[INFO]	[3155]	[build/release64/sls/ilogtail/elogtail.cpp:123]	change working
dir:/usr/local/ilogtail/				
[2018-09-13 01:13:59.025443]	[INFO]	[3155]	[build/release64/sls/ilogtail/AppConfig.cpp:175]	load logtail config file, path:/etc/i
logtail/conf/ap-southeast-2/i	logtail_con	fig.json		
[2018-09-13 01:13:59.025460]	[INFO]	[3155]	[build/release64/sls/ilogtail/AppConfig.cpp:176]	load logtail config file, detail:{
"config_server_address" :	"http://log	tail.ap-so	utheast-2-intranet.log.aliyuncs.com",	
"data_server_list" : [
{				
"cluster" : "ap-sout	heast-2",			
"endpoint" : "ap-sou	theast-2-in	tranet.log	.aliyuncs.com"	
}				
]				

Logtail plug-in log file (logtail_plugin.LOG)

The logtail_plugin.LOG file records the operational logs of Logtail plug-ins. The levels of logs in ascending order include INFO, WARN, ERROR. You can ignore logs at the INFO level.

If an exception such as CANAL_RUNTIME_ALARM occurs, you can troubleshoot the exception based on the logtail_plugin.LOG file.

• File path

• Linux: The file is stored in /usr/local/ilogtail/logtail_plugin.LOG.

- Windows:
 - 64-bit: The file is stored in C:\Program Files (x86)\Alibaba\Logtail\logtail_plugin.LOG.
- 32-bit: The file is stored in C:\Program Files\Alibaba\Logtail\logtail_plugin.LOG.
- Containers: The file is stored in /usr/local/ilogtail/logtail_plugin.LOG.
- Sample file

If you run the tail /usr/local/ilogtail/logtail_plugin.LOG command, the following output is returned:

2018-09-13 02:55:30 [INF] [docker_center.go:525] [func1] docker fetch all:start 2018-09-13 02:55:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop 2018-09-13 03:00:30 [INF] [docker center.go:525] [func1] docker fetch all:start 2018-09-13 03:00:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop 2018-09-13 03:03:26 [INF] [log_file_reader.go:221] [ReadOpen] [##1.0##sls-zc-test-hz-pub\$docker-stdout-config,k8s-stdout] open file for read. offset:40379573 6f1148b2e2f31bd3410f5b2d624-json.log status:794354-64769-40379963 2018-09-13 03:03:26 [INF] [log_file_reader.go:221] [ReadOpen] [##1.0##k8s-log-c12ba2028cfb444238cd9ac1286939f0b\$docker-stdout-config,k8s-std out] open file for read, 6f1148b2e2f31bd3410f5b2d624-json.log offset:40379573 status:794354-64769-40379963 2018-09-13 03:04:26 [INF] [log file reader.go:308] [CloseFile] [##1.0###sls-zc-test-hz-pub\$docker-stdout-config,k8s-stdout] close file, reason:no read timeout 6f1148b2e2f31bd3410f5b2d624-json.log offset:40379963 status:794354-64769-40379963 2018-09-13 03:04:27 [INF] [log_file_reader.go:308] [CloseFile] [##1.0###k8s-log-c12ba2028cfb444238cd9ac1286939f0b\$docker-stdout-config,k8s-st dout] close file, reason:no read timeout 2018-09-13 03:05:30 [INF] [docker_center.go:525] [func1] docker fetch all:start 2018-09-13 03:05:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop

Container path mapping file (docker_path_config.json)

The docker_path_config.json file is created only when you collect container logs. The file records the path mappings between container log files and host log files. The file is in the JSON format.

If the **DOCKER_FILE_MAPPING_ALARM** message appears when you troubleshoot a log collection exception, Docker files fail to be mapped to host files. You can use the docker_path_config.json file to troubleshoot the exception.

? Note

This file is an information record file. Modifications to this file do not take effect. If you delete this file, another file is automatically created without service interruptions.

- File path
 - /usr/local/ilogtail/docker_path_config.json

Sample file

If you run the cat /usr/local/ilogtail/docker_path_config.json command, the following output is returned:

```
{
  "detail" : [
     {
       "config_name" : "##1.0##k8s-log-c12ba2028cfb444238cd9ac1286939f0b$nginx",
        "container_id" : "df19c06e854a0725ea7fca7e0378b0450f7bd3122f94fe3e754d8483fd330d10",
        "params": "{\n \"ID\": \"df19c06e854a0725ea7fca7e0378b0450f7bd3122f94fe3e754d8483fd330d10\",\n \"Path\":
\"/logtail host/var/lib/docker/overlay2/947db346695a1f65e63e582ecfd10ae1f57019a1b99260b6c83d00fcd1892874/diff/var/log\",\n \"Tags\" : [\n
                                       \"access-log\",\n
                                                             \"registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest
\"nginx-type\",\n
       \"_container_name_\",\n \"nginx-log-demo\",\n
\",\n
                                                                                                           \" namespace \",\n
               \"_pod_uid_\",\n
\"default\".\n
                                    \"87e56ac3-b65b-11e8-b172-00163f008685\",\n
                                                                               \"\_container\_ip\_\", \n
                                                                                                      \"172.20.4.224\",\n
                \t = 1 n 
\"purpose\",\n
   "version" : "0.1.0"
}
```

4.3.1.2. Installation

4.3.1.2.1. Install Logtail on a Linux server

This topic describes how to install, update, and uninstall Logtail on a Linux server.

Supported operating systems

You can install Logtail on servers that run one of the following x86-64 Linux operating systems:

- Aliyun Linux 2
- Red Hat Enterprise Linux 6, Red Hat Enterprise Linux 7, and Red Hat Enterprise Linux 8
- CentOS Linux 6, CentOS Linux 7, and CentOS Linux 8
- Debian GNU/Linux 8, Debian GNU/Linux 9, and Debian GNU/Linux 10
- Ubuntu 14.04, Ubuntu 16.04, Ubuntu 18.04, and Ubuntu 20.04
- SUSE Linux Enterprise Server 11, SUSE Linux Enterprise Server 12, and SUSE Linux Enterprise Server 15
- openSUSE Leap 15.1, openSUSE Leap 15.2, and openSUSE Leap 42.3
- · Linux operating systems based on GNU C Library version 2.5 or later

Procedure

? Note

If you run the installation command on a server on which Logtail is installed, the installer uninstalls Logtail from the server, deletes the /usr/local/ilogtail directory, and then reinstalls Logtail. If the installation is successful, Logtail automatically runs and is added as a startup program.

1. Run the following command to download the Logtail installer:

wget http://\${service:sls-backend-server:sls_data.endpoint}/logtail.sh -0 logtail.sh; chmod 755 logtail.sh

? Note

You must replace \${service:sls-backend-server:sls_data.endpoint} in the command with the actual endpoint. You can view the endpoint information on the Project Overview page.

2. Run the installation command.

Start a shell terminal and run the following command as an administrator to install Logtail:

./logtail.sh install

View the version of Logtail

 Open the
 /usr/local/ilogtail/app_info.json
 file. The value of the
 logtail_version
 field is the version of Logtail. Run the
 cat

 /usr/local/ilogtail/app_info.json
 command to view the version of Logtail. The following output is returned:
 cat

```
{
    "UUID" : "0DF18E97-0F2D-486F-B77F-*******,
    "hostname" : "david*******,
    "instance_id" : "F4FAFADA-F1D7-11E7-846C-00163E30349E_*********_1515129548",
    "ip" : "*********",
    "logtail_version" : "0.16.0",
    "os" : "Linux; 2.6.32-220.23.2.ali113.el5.x86_64; #1 SMP Thu Jul 4 20:09:15 CST 2013; x86_64",
    "update_time" : "2018-01-05 13:19:08"
}
```

Update Logtail

You can use the Logtail installer logtail.sh to update Logtail. The installer automatically selects an update method based on the configurations of Logtail that is installed.

? Note

```
Logtail is temporarily stopped during the update. Some files are overwritten. The configuration files and the checkpoint files are retained. No log data is lost during the update.
```

Run the following command to upgrade Logtail:

```
# Download the installer.
wget http://${service:sls-backend-server:sls_data.endpoint}/logtail.sh -0 logtail.sh;
chmod 755 logtail.sh
# Run the update command.
sudo ./logtail.sh upgrade
```

Output:

```
# The update is successful.
Stop logtail successfully.
ilogtail is running
Upgrade logtail success
{
    "UUID": "***",
    "hostname": "***",
    "instance_id": "***",
    "ingtail_version": "0.16.11",
    "os": "Linux; 3.10.0-693.2.2.e17.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
    "update_time": "2018-08-29 15:01:36"
}
# The update fails because the current version is the latest version.
[Error]: Already up to date.
```

Start and stop Logtail

```
    Start Logtail
Run the
/etc/init.d/ilogtaild start
command as an administrator.
    Stop Logtail
Run the
```

/etc/init.d/ilogtaild stop

command as an administrator.

Uninstall Logtail

Start a shell terminal and run the following command as an administrator to uninstall Logtail:

wget http://\${service:sls-backend-server:sls_data.endpoint}/logtail.sh -0 logtail.sh chmod 755 logtail.sh ./logtail.sh uninstall

Install a second Logtail on a machine

When you install a second Logtail on a machine, conflicts occur because the default installation path to the configuration file of the second Logtail is the same as the installation path to the configuration file of the existing Logtail. To resolve the conflicts, you can add the **_public** parameter to the Logtail installation command. This helps distinguish between the two installation paths.

For example, when you install a second Logtail, you can add the **_public** parameter to specify the installation path as /usr/local/ilogtail_public .

1. Log on to the Apsara Infrastructure Management console.

2. Obtain the endpoint that you can use to install a second Logtail.

i. In the left-side navigation pane, choose **O&M Tools > Basic Data Management**.

- ii. On the Basic Data Management page, click Service Registration Variable Dashboard.
- iii. On the Service Registration Variable Dashboard page, search for sls in the Service column.
- iv. Find the sls-backend-server service, and click the value of Service Registration column to obtain the value of the sls_data.endpoint variable. The value indicates the endpoint that you can use.

"sls_data.endpoint": "data.cn-qingdao-env66-d01.sls-pub.inter.env66.shuguang.com",

3. Install Logtail on your machine.

i. Run the following command on your machine to download the Logtail installer:

wget http://\${service:sls-backend-server:sls_data.endpoint}/logtail.sh -0 logtail.sh; chmod 755 logtail.sh

```
? Note
```

You must replace \${service:sls-backend-server:sls_data.endpoint} in the command with the endpoint that is obtained.

ii. Run the installation command on your machine.

Start a shell terminal and run the following command as an administrator to install Logtail:

./logtail.sh install -s _public

The -s parameter specifies a Logtail installation path in /usr/local/. If you do not add the -s parameter, the Logtail installation path is /usr/local/ilogtail . After you run the command, the Logtail installation path is /usr/local/ilogtail_public .

4. View the status of the second Logtail. Run the following command:

/etc/init.d/ilogtaild_public status

If the status of Logtail is Running after the installation is complete, you do not need to restart Logtail. Otherwise, run the following command to restart Logtail: /etc/init.d/ilogtaild_public stop && /etc/init.d/ilogtaild_public start

4.3.1.2.2. Install Logtail on a Windows server

This topic describes how to install Logtail on a Windows server.

Prerequisites

At least one Windows server is available.

Supported operating systems

The following Windows operating systems are supported.

? Note

Logtail supports Windows Server 2008 and Windows 7 that run on x86 or x86_64 and other Windows operating systems that run on x86_64

- Microsoft Windows Server 2008
- Microsoft Windows Server 2012
- Microsoft Windows Server 2016
- Microsoft Windows Server 2019
- Microsoft Windows 7
- Microsoft Windows 10
- Microsoft Windows Server Version 1909
- Microsoft Windows Server Version 2004

Install Logtail

- 1. Download the installation package.
 - If the server resides in a region in the Chinese mainland, click Logtail installation package.
- If the server resides in a region outside the Chinese mainland, click Logtail installation package.
- 2. Decompress the logtail_installer.zip package to the current directory.

3. Run an installation command.

Run Windows PowerShell or Command Prompt as an administrator. Go to the logtail_installer directory and run the installation command based on the network type.

./logtail_installer.exe. install \${region}

```
ONOTE
You must replace s{region} in the command with the actual endpoint. For more information about endpoints, see View the endpoint of a project.
```

Installation path

After you run the installation command, Logtail is automatically installed. The installation path varies based on the operating system. You cannot change the installation path.

• 32-bit Windows: C:\Program Files\Alibaba\Logtail

• 64-bit Windows: C:\Program Files (x86)\Alibaba\Logtail

? Note

You can run 32-bit and 64-bit applications in a 64-bit Windows operating system. To ensure compatibility, the operating system stores 32-bit applications in a separate x86 directory.

Logtail for Windows is a 32-bit application and is installed in the Program Files (x86) directory in a 64-bit Windows operating system. If Logtail for 64-bit Windows is available, Logtail is automatically installed in the Program Files directory.

View the version of Logtail

Go to the installation directory and open the app_info.json file. The value of the logtail_version field in the file is the Logtail version.

In the following example, the Logtail version is V1.0.0.0.

```
{
    "logtail_version" : "1.0.0.0"
}
```

Update Logtail

Automatic update

In most cases, Logtail on Windows servers supports automatic update. However, Logtail whose version is earlier than V1.0.0.0 can only be manually updated to Logtail V1.0.0.0 or later.

Manual update

To manually update Logtail, you must download and decompress the most recent installation package. Then, you can install Logtail as prompted to complete the update. For more information, see Install Logtail.

? Note

During manual update, Logtail is automatically uninstalled and then reinstalled. The original files in the installation directory are deleted. Before you manually update Logtail, we recommend that you back up the files.

Start and stop Logtail

1. Choose Start Menu > Control Panel > Administrative Tools > Services.

- 2. In the Services dialog box, select the service that you want to manage.
- For Logtail V0.x.x.x, select LogtailWorker.
- For Logtail V1.0.0.0 or later, select LogtailDaemon.

3. Right-click the service and select Start, Stop, or Restart.

Uninstall Logtail

Run Windows PowerShell or Command Prompt as an administrator. Go to the logtail_installer directory and run the following command to uninstall Logtail. The directory contains the files that are extracted from the installation package.

.\logtail_installer.exe uninstall

After Logtail is uninstalled, the Logtail installation directory is deleted. However, some configuration files are retained in the C:\LogtailData directory. You can manually delete the files based on your business requirements. The following configuration files are retained:

- checkpoint: contains information about the checkpoints that are generated by Logtail plug-ins. This file is generated only if the Logtail plug-ins are used.
- user_config.d: contains local collection configurations.
 JSON files in the directories are considered collection configurations. For example, /usr/local/ilogtail/user_log_config.json is considered a collection configuration.
- logtail_check_point: contains information about the checkpoints that are generated by Logtail.
- users: contains the user identifier files that are configured.

4.3.1.2.3. Install Logtail components in a Kubernetes cluster

This topic describes how to install Logtail components in a Kubernetes cluster.

Background information

Before you can collect container logs from a Kubernetes cluster, you must install Logtail components. When you install Logtail components, the following operations are automatically complete:

- 1. The alibaba-log-configuration ConfigMap is created. This ConfigMap stores the configuration information about Log Service, such as project information.
- 2. Optional. The AliyunLogConfig custom resource definition (CRD) is created.
- 3. Optional. The alibaba-log-controller Deployment is created. This Deployment is used to monitor the changes in the AliyunLogConfig CRD and create Logtail configurations.
- 4. The logtail-ds DaemonSet is created. This DaemonSet is used to collect logs from nodes.

Container Service for Kubernetes clusters

You can install Logtail components in an existing Container Service for Kubernetes cluster. You can also install Logtail components when you create a Container Service for Kubernetes cluster. To install Logtail components when you create a Container Service for Kubernetes cluster, you must select **Enable Log Service**.

Install Logtail components in an existing Container Service for Kubernetes cluster

() Important

If your Container Service for Kubernetes cluster is a dedicated Kubernetes cluster or a managed Kubernetes cluster, you can follow the instructions in this section to install Logtail components in your Container Service for Kubernetes cluster.

- 1. Log on to the server where Container Service for Kubernetes is deployed.
- 2. In the left-side navigation pane, click Clusters.
- 3. On the **Clusters** page, find and click the cluster in which you want to install Logtail components.
- 4. In the left-side navigation pane of the page that appears, choose Operations > Add-ons.
- 5. On the Logs and Monitoring tab, find the logtail-ds component and click Install.

After logtail-ds is installed, Log Service automatically creates a project named k8s-log -\${your_k8s_cluster_id} , a machine group named k8sgroup-\${your_k8s_cluster_id} , and a Logstore named config-operation-log in the project. k8s-log -\${your_k8s_cluster_id} , a machine group named k8s-

() Important Do not delete the config-operation-log Logstore.

Install Logtail components when you create a Container Service for Kubernetes cluster

- 1. Log on to the server where Container Service for Kubernetes is deployed.
- 2. In the left-side navigation pane, click **Clusters**.
- 3. On the Clusters page, click Create Kubernetes Cluster.
- 4. In the Component Configurations step, select Enable Log Service.
 - ? Note

In this example, only the steps that are required to enable Log Service are provided. For more information about how to create a cluster, see Create a Kubernetes cluster in Container Service for Kubernetes User Guide.

If you select **Enable Log Service**, the system prompts you to create a Log Service project. For more information about the log management structure of Log Service, see **Project in the Quick start topic of Log Service User Guide**. You can use one of the following methods to create a project:

• Select Project

You can select an existing project to manage the container logs that are collected.

• Create Project

Log Service automatically creates a project named k@s-log-{ClusterID} to manage the container logs that are collected. ClusterID indicates the unique ID of the Container Service for Kubernetes cluster that is created.

After the Logtail components are installed, a machine group named k8s-group-\${your_k8s_cluster_id} and a Logstore named config-operationlog are automatically created in your project.

① Important Do not delete the config-operation-log Logstore.

Self-managed Kubernetes clusters

- 1. Log on to the server where Container Service for Kubernetes is deployed.
- 2. Create a project whose name starts with k8s-log-custom- .
 - Example: k8s-log-custom-sd89ehdq. For more information, see Manage a project.
- 3. Log on to your Kubernetes cluster.
- 4. Run the following commands to install Logtail and dependent components.
 - () Important
 - Make sure that the kubectl command-line tool is installed on the machine on which you want to run the commands.
 - alibaba-log-controller is available only in Kubernetes 1.6 or later.
 - If you no longer need to use CRDs, you can delete the alibaba-cloud-log/templates/alicloud-log-config.yaml file and rerun the following commands. If the ./alicloud-log-k8s-custom-install.sh: line 111: /root/alibaba-cloud-log/templates/alicloud-log-crd.yaml: No such file or directory error message appears, you can ignore the error.
 - i. Download the installation script.

wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/kubernetes/alicloud-log-k8s-custom-install.sh

ii. Modify permissions to limit access to the installation script.

chmod 744 ./alicloud-log-k8s-custom-install.sh

iii. Install Logtail and dependent components.

sh ./alicloud-log-k8s-custom-install.sh your-project-suffix region-id aliuid access-key-id access-key-secret

The following table describes the parameters that are included in the preceding command. You can configure the parameters based on your business requirements.

Parameter	Description
your-project-suffix	The part that is specified after k8s-log-custom- in the name of your project. Use the project that you created in Self-managed Kubernetes clusters. For example, if the project name is k8s-log-custom-sd89ehdg , set the value to sd89ehdq .
region-id	The ID of the region where your project resides. For example, the ID of the China (Hangzhou) region is cn- hangzhou . For more information, see Obtain the endpoint of Log Service in Log Service Developer Guide .
aliuid	The ID of your Apsara Stack tenant account. For more information, seeConfigure a user identifier.
access-key-id	The AccessKey ID of your Apsara Stack tenant account. We recommend that you use the AccessKey pair of a RAM user and attach the AliyunLogFullAccess policy to the RAM user. For more information, see Create a RAM role.
access-key-secret	The AccessKey secret of your Apsara Stack tenant account. We recommend that you use the AccessKey pair of a RAM user and attach the AliyunLogFullAccess policy to the RAM user. For more information, see Create a RAM role and Grant permissions to a RAM role.

After Logtail and the components are installed, a machine group named k8s-group-\${your_k8s_cluster_id} and a Logstore named configoperation-log are automatically created in your project.

() Important

- Do not delete the config-operation-log Logstore.
- If you install Logtail components in a self-managed Kubernetes cluster, Logtail is granted the privileged permissions. This helps prevent the container text file busy error that occurs when other pods are deleted. For more information, see Bug 1468249, Bug 1441737, and Issue 34538.

FAQ

• How do I view the version of a container image?

You can view the version of a container image in an image repository. For more information, visit the **logtail-ds image repository** or the **alibaba-log-controller image repository** page.

How do I upgrade Logtail components?

You can upgrade Logtail components in automatic or manual mode. For more information, see Upgrade Logtail components in a Kubernetes cluster. • How do I collect container logs from multiple Kubernetes clusters to the same Log Service project?

Container Service for Kubernetes clusters

If you want to collect container logs from multiple Container Service for Kubernetes clusters to the same Log Service project, you must select the same project when you create the Container Service for Kubernetes clusters.

Self-managed Kubernetes clusters

If you want to collect container logs from multiple self-managed Kubernetes clusters to the same Log Service project, you must set the **{your-project-suffix}** parameter to the same value when you install Logtail components in each of the Kubernetes clusters.

? Note

You can collect container logs from multiple self-managed Kubernetes clusters to the same Log Service project only if the Kubernetes clusters reside in the same region.

• How do I view the logs of Logtail?

The logs of Logtail are stored in the files named ilogtail.LOG and logtail_plugin.LOG in the /usr/local/ilogtail/ directory of a Logtail container. The stdout and stderr of the Logtail container are not for reference. You can ignore the following stdout and stderr:

```
start umount useless mount points, /shm$|/merged$|/mqueu$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c37869500fbe2bdb95d13ble110172ef57fe840c82155/merged: must be superuser to
umount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa1939992755de1f85d25009528daa749clbf8c16edff44beab6e69718/merged: must be superuser to
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6880dc4e8a640ble16c22dbe/merged: must be superuser to
umount
.....
xargs: mount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
ilogtail is running
```

Cloud Defined Storage

 How do I view the status of Log Service components in Kubernetes clusters? Run the following commands:

kubectl get deploy alibaba-log-controller -n kube-system kubectl get ds logtail-ds -n kube-system

• What do I do if alibaba-log-controller fails to start?

Check whether alibaba-log-controller is installed by using the following method:

- Run the installation command on the control plane of your Kubernetes cluster.
- Specify the ID of your Kubernetes cluster in the installation command.

If alibaba-log-controller is not installed by using the preceding method, run the kubectl delete -f deploy command to delete the installation template that is generated. Then, run the installation command again.

- How do I view the status of the Logtail DaemonSet in a Kubernetes cluster?
- Run the kubectl get ds -n kube-system command to view the status of the Logtail DaemonSet.

?	Note
<u> </u>	

The default namespace to which a Logtail container belongs is kube-system.

• How do I view the version number, IP address, startup time, and status of Logtail?

• Run the following command to view the status of Logtail:

kubectl get po -n kube-system | grep logtail

The following output is returned:

NAME	READY	STATUS	RESTARTS	AGE
logtail-ds-gb92	k 1/1	Running	J O	2h
logtail-ds-wm71	w 1/1	Running	J 0	4d

• Run the following command to view the version number and IP address of Logtail:

kubectl exec logtail-ds-gb92k -n kube-system cat /usr/local/ilogtail/app_info.json

The following output is returned:

```
"UUID" : "",
"hostname" : "logtail-ds-gb92k",
"instance_id" : "0EBB2B0E-0A3B-11E8-B0CE-0A58AC140402_172.20.4.2_1517810940",
"ip" : "192.0.2.0",
"logtail_version" : "0.16.2",
"os" : "Linux; 3.10.0-693.2.2.e17.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
"update_time" : "2021-02-05 06:09:01"
```

• How do I view the operational logs of Logtail?

The operational logs of Logtail are stored in the ilogtail.LOG file in the /usr/local/ilogtail/ directory. If the log file is rotated, the generated files are compressed and stored as ilogtail.LOG.x.gz.

Examples:

```
[root@iZbpldsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-ds-gb92k -n kube-system tail /usr/local/ilogtail/ilogtail.LOG
[2018-02-05 06:09:02.168693] [INFO] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:104] logtail plugin Resume:start
[2018-02-05 06:09:02.168807] [INFO] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:106] logtail plugin Resume:success
[2018-02-05 06:09:02.168822] [INFO] [9] [build/release64/sls/ilogtail/EventDispatcher.cpp:369] start add existed check point events, size:0
[2018-02-05 06:09:02.168827] [INFO] [9] [build/release64/sls/ilogtail/EventDispatcher.cpp:511] add existed check point events, size:0 cache
size:0 event size:0 success count:0
```

• How do I restart Logtail for a pod?

i. Stop Logtail.

In the following command, logtail-ds-gb92k -n specifies the container, and kube-system specifies the namespace. Configure the parameters based on your business requirements.

kubectl exec logtail-ds-gb92k -n kube-system /etc/init.d/ilogtaild stop

If the following output is returned, Logtail is stopped:

```
kill process Name: ilogtail pid: 7
kill process Name: ilogtail pid: 9
stop success
```

ii. Start Logtail.

In the following command, logtail-ds-gb92k -n specifies the container, and kube-system specifies the namespace. Configure the parameters based on your business requirements.

kubectl exec logtail-ds-gb92k -n kube-system /etc/init.d/ilogtaild start

If the following output is returned, Logtail is started:

ilogtail is running

- How do I collect the logs of control-plane components?
- To collect the logs of control-plane components from a managed Container Service for Kubernetes cluster, you can enable the log collection feature in the Container Service for Kubernetes console. For more information, see Collect the logs of control-plane components in managed clusters in Container Service for Kubernetes User Guide.

 For more information about how to collect logs from self-managed Kubernetes clusters and dedicated Container Service for Kubernetes clusters, see Use the Log Service console to collect container stdout and stderr in DaemonSet mode.

What to do next

Create Logtail configurations to collect container logs.

- DaemonSet mode
 - For more information about how to collect container logs by using CRDs, see Use CRDs to collect container logs in DaemonSet mode.
 - For more information about how to collect container stdout and stderr by using the Log Service console, see Use the Log Service console to collect container stdout and stderr in DaemonSet mode.
 - For more information about how to collect container text logs by using the console, see Use Log Service to collect log data from containers in Container Service for Kubernetes User Guide.
- Sidecar mode
 - For more information about how to collect container text logs by using CRDs, see Use CRDs to collect container text logs in Sidecar mode.
 - For more information about how to collect container text logs by using the Log Service console, see Use the Log Service console to collect container text logs in Sidecar mode.

4.3.1.2.4. Upgrade Logtail components in a Kubernetes cluster

This topic describes how to upgrade Logtail components.

Upgrade description

We recommend that you use the automatic upgrade method in common scenarios. If you have modified parameters such as environment variables in the logtail-ds DaemonSet or the alibaba-log-controller Deployment, we recommend that you use the manual upgrade method to retain your modifications.

Before you upgrade the Logtail components, we recommend that you back up the description files that are related to the Logtail components. For more information, see Appendix: Backup and rollback.

() Important

An upgrade requires a few seconds to complete. During an upgrade, the Logtail container is restarted, which may cause a small amount of data to be collected again or lost during collection.

Automatic upgrade

! Important

If you use the automatic upgrade method, your modifications to the parameters in the logtail-ds DaemonSet and alibaba-log-controller Deployment are not retained.

Container Service for Kubernetes clusters

- 1. Log on to the server where Container Service for Kubernetes is deployed.
- 2. In the left-side navigation pane, click Clusters.
- 3. On the **Clusters** page, find and click the cluster that you want to manage.
- 4. In the left-side navigation pane of the page that appears, choose **Operations** > **Add-ons**.
- 5. On the Logs and Monitoring tab, find logtail-ds and click Upgrade
- 6. In the Update dialog box, click OK.

() Important

If the component cannot be upgraded to the most recent Logtail version, the Kubernetes version of your cluster is outdated. In this case, you must upgrade the Kubernetes version of your cluster first or use the manual upgrade method.

After the upgrade is performed, you can view the status of each logtail-ds pod in the Container Service for Kubernetes console. If each logtail-ds pod is in the running state, the upgrade is successful.

Self-managed Kubernetes clusters

You can reinstall the Logtail components to complete the automatic upgrade. For more information, see Self-managed Kubernetes clusters.

Manual upgrade

```
() Important
```

If you use the manual upgrade method, your existing configurations are not updated, and some features may not be supported.

Container Service for Kubernetes clusters

A manual upgrade covers both logtail-ds and alibaba-log-controller. In most cases, you need to only upgrade logtail-ds to obtain the collection capabilities provided in the most recent version of Logtail. If you want to obtain the custom resource definition (CRD)-based collection capabilities provided in the most recent version of Logtail, you must also upgrade alibaba-log-controller. The following procedure shows how to upgrade logtail-ds.

1. Log on to the server where Container Service for Kubernetes is deployed.

- 2. In the left-side navigation pane, click Clusters.
- 3. On the **Clusters** page, find and click the cluster that you want to manage.
- 4. In the left-side navigation pane, choose Workloads > DaemonSets.

? Note

If you want to upgrade alibaba-log-controller, choose **Workloads > Deployments**. Then, set Namespace to **kube-system** and find alibaba-log-controller.

- 5. Set Namespace to kube-system. Then, find logtail-ds and click Edit in the Actions column.
- 6. Check whether the required environment variables exist.
- If the ALIYUN LOGTAIL_CONFIG, ALIYUN_LOGTAIL_USER_ID, or ALIYUN_LOGTAIL_USER_DEFINED_ID environment variable does not exist, your Logtail version may be outdated. You can submit a **ticket** to consult for Logtail upgrade methods.
- 7. Click Select Image Version to the right of Image Version.
- 8. In the Image Version dialog box, click the most recent version and click OK.
- 9. In the right-side pane of the page that appears, click Update

After the upgrade is performed, you can view the status of each logical-ds pod in the Container Service for Kubernetes console. If each logical-ds pod is in the running state, the upgrade is successful.

Self-managed Kubernetes clusters

- 1. Log on to your Kubernetes cluster.
- 2. Upgrade logtail-ds.

i. Run the following command to enter the configuration mode:

kubectl edit ds -n kube-system logtail-ds

ii. Modify the image field.

You can view the most recent image on the logtail page.

After the upgrade is performed, you can run the kubectl describe -n kube-system ds logtail-ds | grep Status: command to view the status of each logtail-ds pod. If each logtail-ds pod is in the running state, the upgrade is successful.

- 3. Upgrade alibaba-log-controller.
 - i. Run the following command to enter the configuration mode:

kubectl edit deployment -n kube-system alibaba-log-controller

ii. Modify the image field.

You can obtain the most recent image on the alibabacloud-log-controller page.

After the upgrade is performed, you can run the kubectl describe -n kube-system deployment alibaba-log-controller | grep Replicas: command to view the status of each alibaba-log-controller pod. If each alibaba-log-controller pod is in the updated state, the upgrade is successful.

Appendix: Backup and rollback

Backup

Before you upgrade the Logtail components, we recommend that you back up the description files that are related to the Logtail components. Example:

```
kubectl get ds -n kube-system logtail-ds -o yaml > logtail-ds.yaml
kubectl get deployment -n kube-system alibaba-log-controller -o yaml > alibaba-log-controller.yaml
kubectl get crd aliyunlogconfigs.log.alibabacloud.com -o yaml > aliyunlogconfigs-crd.yaml
kubectl get cm -n kube-system alibaba-log-configuration -o yaml > alibaba-log-configuration.yaml
```

Rollback

The following procedure shows how to roll back to a version.

Note

The YAML files that are backed up before an upgrade contains redundant information. You must manually delete the redundant information before you can restore the configurations of Logtail. You can use the kubectl-neat tool to delete the redundant information. You must delete the following fields: metadata.creationTimestamp, metadata.generation, metadata.resourceVersion, metadata.uid, and status.

1. Delete redundant information from the backup files.

- cat logtail-ds.yaml | kubectl-neat > neat-logtail-ds.yaml
- cat alibaba-log-controller.yaml | kubectl-neat > neat-alibaba-log-controller.yaml
 cat aliyunlogconfigs-crd.yaml | kubectl-neat > neat-aliyunlogconfigs-crd.yaml
- cat alibaba-log-configuration.yaml | kubectl-neat > neat-alibaba-log-configuration.yaml

2. Use the backup files after the deletion to restore the configurations of Logial.

```
kubectl apply -f neat-logtail-ds.yaml
```

```
kubectl apply -f neat-alibaba-log-controller.yaml
```

- kubectl apply -f neat-aliyunlogconfigs-crd.ya kubectl apply -f neat-alibaba-log-configuration.yaml

4.3.1.2.5. Configure the startup parameters of Logtail

Log Service limits the collection performance of Logtail to prevent Logtail from consuming excessive server resources. If Logtail consumes excessive server resources, other services on the server may be affected. If you want to improve the collection performance of Logtail, you can modify the startup parameters of Logtail.

Scenarios

You can modify the startup parameters of Logtail in the following scenarios:

- You need to collect logs from a large number of log files and the log files occupy a large amount of memory. For example, you need to collect logs from more than 100 files or the log monitoring directory contains more than 5,000 log files.
- Log data is transmitted at a high speed, which causes high CPU utilization. For example, Logtail collects log data at a speed that exceeds 2 MB/s in simple mode and at a speed that exceeds 1 MB/s in full regex mode.
- Logitail sends data to Log Service at a speed that exceeds 10 MB/s

Recommended parameter values

If you want to collect logs from JSON files, you can use the following parameter values that are obtained from real-world practice. The collection performance of Logtail in full regex mode and in delimiter mode is similar to the collection performance of Logtail in JSON mode. The collection performance of Logtail in simple mode is five times higher than the collection performance of Logtail in JSON mode. Both the complexity of data and rules and the numbers of directories and files from which you want to collect logs affect CPU utilization and memory usage. We recommend that you configure the following parameters based on the values in the table and your business requirements.

Host environment

Parameter	Default collection speed	Collection speed higher than 10 MB/s	Collection speed higher than 20 MB/s	Collection speed higher than 40 MB/s
cpu_usage_limit	0.4	1	2	4
mem_usage_limit	384	1024	2048	4096
max-bytes-per-sec	20971520	209715200	209715200	209715200
process_thread_count	1	2	4	8
send_request_concurrency	4	20	40	80

• Container or Kubernetes environment

Environment variable	Default collection speed	Collection speed higher than 10 MB/s	Collection speed higher than 20 MB/s	Collection speed higher than 40 MB/s
cpu_usage_limit	2	3	5	9
mem_usage_limit	2048	2048	2048	4096
max_bytes_per_sec	209715200	209715200	209715200	209715200
process_thread_count	1	2	4	8
send_request_concurrency	20	20	40	80
resources.limits.cpu	500M	1000M	2000M	4000M
resources.limits.memory	2 Gi	2 Gi	3 Gi	5 Gi

If you want to collect logs from a container or a Kubernetes cluster, you can modify the startup parameters of Logtail by modifying DaemonSetrelated environment variables. ConfigMaps are referenced by some environment variables, and the path to the ConfigMaps is **configmap > kube**system > alibaba-log-configuration. You can also modify resources.limits.cpu and resources.limits.memory in daemonset > kubesystem > logtail-ds to prevent the excessive usage of container resources.

If you configure the Logtail startup parameters based on the values of the **Collection speed higher than 40 MB/s** column in the preceding tables, the collection performance of Logtail approaches the upper limit. In this case, the performance does not significantly improve even if more threads are created. The following table describes the upper limit of the collection performance that Logtail can deliver in different collection modes.

? Note

The actual collection performance may vary based on the test environment and the production environment.

Collection mode	Upper limit
Simple mode	440 MB/s
Full regex mode	70 MB/s
Delimiter mode	75 MB/s
JSON mode	75 MB/s

Configure the startup parameters of Logtail

1. Open the /usr/local/ilogtail/ilogtail_config.json file on the server on which Logtail is installed.

If you want to collect logs from a host, you can perform this step to configure the startup parameters of Logtail.

If you want to collect logs from a container or a Kubernetes cluster, you can modify the startup parameters of Logtail by modifying DaemonSetrelated environment variables. ConfigMaps are referenced by some environment variables, and the path to the ConfigMaps is **configmap > kube**system > alibaba-log-configuration.

2. Configure the startup parameters of Logtail based on your business requirements.

The following example shows the startup parameters of Logtail:

	{	
		"cpu_usage_limit" : 0.4,
		"mem_usage_limit" : 384,
		"max_bytes_per_sec" : 20971520,
		"process_thread_count" : 1,
		"send_request_concurrency" : 4,
		"buffer_file_num" : 25,
		"buffer_file_size" : 20971520,
		"buffer_file_path" : "",
	}	
T		

? Note

- The following table describes the commonly used startup parameters of Logtail. You can retain the default values for other startup parameters.
- $\circ\;$ You can add or modify startup parameters based on your business requirements.

The following table describes the startup parameters of Logtail.

Table 1. Startup parameters of Logtail

Parameter	Туре	Description	Example
cpu_usage_limit	double	 The CPU utilization threshold for Logtail. The calculation is based on a single core. Valid values: 0.1 to the number of CPU cores of the current server Default value: 0.4 Marning cpu_usage_limit specifies a soft limit. The actual CPU utilization of Logtail may exceed the limit. If the CPU utilization of Logtail remains higher than this limit for 5 minutes, the system triggers a circuit breaker. Then, Logtail automatically restarts. For example, you set the parameter to 0.4. If the CPU utilization of Logtail remains higher than 40% for 5 minutes based on a single core, Logtail automatically restarts. In most cases, a single core supports a collection speed of about 100 MB/s in simple mode and about 20 MB/s in full regex mode. 	"cpu_usage_limit" : 0.4
mem_usage_limit	int	 The memory usage threshold for Logtail. Unit: MB. Valid values: 128 to 8192 Default value: 384 for a host environment and 2048 for Container Service for Kubernetes (ACK) components Marning mem_usage_limit specifies a soft limit. The actual memory usage of Logtail may exceed the limit. If the memory usage of Logtail remains higher than this limit for 5 minutes, the system triggers a circuit breaker. Then, Logtail automatically restarts. The collection speed, monitored directories, number of log files, and number of synchronously sent requests are related to the mem_usage_limit parameter. For more information, see Limits. 	"mem_usage_limit" : 384
max_bytes_per_sec	int	The highest speed at which Logtail sends raw data. Unit: bytes per second. • Valid values: 1024 to 52428800 • Default value: 20971520 For example, if you set the parameter to 2097152 , the highest speed at which Logtail sends data is 2 MB/s.	"max_bytes_per_sec" : 2097152
process_thread_count	int	 The number of threads that are used by Logtail to process data. Valid values: 1 to 64 Default value: 1 In most cases, a thread provides a write speed of 24 MB/s in simple mode and 12 MB/s in full regex mode. We recommend that you retain the default value for this parameter. 	"process_thread_coun t" : 1

send_request_concurr ency	int	The maximum number of concurrent requests that can be sent by Logtail to asynchronously send data. • Valid values: 1 to 1000 • Default value: 20 If Log Service provides a high transactions per second (TPS), you can set this parameter to a larger value. Each concurrent request supports a network throughput of 0.5 MB/s to 1 MB/s. The actual network throughput for a concurrent request varies based on the network latency.	"send_request_concur rency" : 4
buffer_file_num	int	 The maximum number of files that can be cached. Valid values: 1 to 100. Default value: 25 If a network error occurs or the limits of data write are reached, Logtail caches parsed logs to the local files in the installation directory. Logtail parses raw logs in real time. After the issues are fixed, Logtail retries to send the cached logs. 	"buffer_file_num" : 25
buffer_file_size	int	 The maximum size of a cached file. Unit: bytes. Valid values: 1048576 to 104857600 Default value: 20971520 The maximum disk space that can be occupied by cached files is calculated by multiplying the value of the buffer_file_size parameter by the value of thebuffer_file_num parameter. 	"buffer_file_size" : 20971520
buffer_file_path	String	The directory in which cached files are stored. This parameter is empty by default, which indicates that cached files are stored in the installation directory of Logtail. The default directory is /usr/local/ilogtail. If you specify a value for this parameter, you must move the cached files whose name matches logtail_buffer_file_* from the installation directory of Logtail to the directory that you specify. This way, Logtail can read, send, and then delete the cached files.	"buffer_file_path" : ""
bind_interface	String	The name of the network interface controller (NIC) that is associated with the server on which Logtail is installed. This parameter is empty by default, which indicates that the server is automatically associated with an available NIC. If you specify a value for this parameter, such as eth1, Logtail uses the NIC to upload logs. This parameter is available only if Logtail runs on a Linux server.	"bind_interface" : ""
check_point_filename	String	The path to the checkpoint files of Logtail. Default value:/tmp/logtail_check_point.	"check_point_filenam e" : /tmp/logtail_check_po int
check_point_dump_int erval	int	The interval at which Logtail updates checkpoint files. Default value: 900. Unit: seconds. If you retain the default value, Logtail updates checkpoint files at 15-minute intervals. This parameter is available only for Logtail V1.0.19 or later.	"check_point_dump_i nterval" : 900
user_config_file_path	String	The path to the file that stores Logtail configurations. The file is nameduser_log_config.json and stored in the directory of the BIN file that is created for the Logtail process.	"user_config_file_path " : user_log_config.json
docker_file_cache_pat h	String	The path to the file that records the path mappings between container files and host files. By default, the path is /usr/local/ilogtail/docker_path_config.json. This parameter is available only for Logtail V0.16.54 or later.	"docker_file_cache_pa th": /usr/local/ilogtail/doc ker_path_config.json
discard_old_data	Boolean	Specifies whether to discard historical logs. Default value: true. This value indicates that logs that were generated more than 12 hours before the current time are discarded.	"discard_old_data" : true
ilogtail_discard_interv al	int	The time threshold for discarding logs. If the difference between the time at which the logs were generated and the current time exceeds the threshold, the logs are discarded. Default value: 43200. Unit: seconds. The value 43200 indicates that the threshold is 12 hours.	"ilogtail_discard_inter val": 43200
working_ip	String	The server IP address that is reported by Logtail to Log Service. This parameter is empty by default, which indicates that Log Service automatically obtains the IP address of the server on which Logtail is installed.	"working_ip" : ""
working_hostname	String	The server hostname that is reported by Logtail to Log Service. This parameter is empty by default, which indicates that Log Service automatically obtains the hostname of the server on which Logtail is installed.	"working_hostname" : ""

max_read_buffer_size	long	The maximum size of a log that Logtail can read. Unit: bytes. Default value: 524288. The default value 524288 indicates that the maximum size is 512 KB. Maximum value: 4194304. The value 4194304 indicates that the maximum size is 4 MB. If the size of a log exceeds 524,288 bytes, you can change the value of this parameter.	"max_read_buffer_siz e" : 524288
oas_connect_timeout	long	The timeout period of the connection that is established by Logtail to send a request to obtain the Logtail configuration or AccessKey pair. Default value: 5. Unit: seconds. If the connections cannot be established before timeout due to poor network conditions, you can change the value of this parameter.	"oas_connect_timeout " : 5
oas_request_timeout	long	The timeout period of the request that is sent by Logtail to obtain the Logtail configuration or AccessKey pair. Default value: 10. Unit: seconds. If the connections cannot be established before timeout due to poor network conditions, you can change the value of this parameter.	"" : 10
data_server_port	long	If you set the data_server_port parameter to 443 , Logtail transfers data to Log Service over HTTPS. This parameter is available only for Logtail V1.0.10 or later.	"data_server_port": 443
enable_log_time_auto _adjust	Boolean	If you set the enable_log_time_auto_adjust parameter to true , the log time is adapted to the local time of the server. To ensure data security, Log Service checks the time information in requests, including the requests sent by Logtail. This information indicates the time at which a request is sent. Log Service rejects requests that are sent 15 minutes earlier or later than the time in Log Service. The time information in a request is considered as the local time of the server. In some test scenarios, the local time must be changed to a future point in time. If you change the local time of Desrvice. You can use this parameter to adapt the log time to the local time of the server. This parameter is available only for Logtail V1.0.19 or later.	"enable_log_time_aut o_adjust": true
		 Important If you set the enable_log_time_auto_adjust parameter to true, the offset between the time in Log Service and the local time of the server is added to the log time. The offset is updated only if a request is rejected by Log Service. Therefore, the time of a log that is queried by Log Service may be different from the time at which the log is written. Part of the logic for Logtail changes based on the incremental increase of the system time. We recommend that you restart Logtail after you change the local time of the server. 	
accept_multi_config	Boolean	Specifies whether to allow Logtail to collect data from the same file by using multiple Logtail configurations. Default value: false. This value indicates that Logtail cannot collect data from the same file by using multiple Logtail configurations. By default, Logtail can use only one Logtail configuration to collect data from a file. If you want to allow Logtail to collect data from a file by using multiple Logtail configuration to collect data from a file. If you want to allow Logtail to collect data from a file by using multiple Logtail configurations, you can set this parameter to true. Each Logtail configuration has an independent collection process. If multiple Logtail configurations are used to collect data from the same file, the CPU utilization and memory usage increase. This parameter is available only for Logtail V0.16.26 or later.	"accept_multi_config" : true
enable_checkpoint_sy nc_write	Boolean	Specifies whether to enable the sync write feature. Default value: false. This value indicates that the sync write feature is disabled. The sync write feature is used together with the ExactlyOnce write feature. After you enable the ExactlyOnce write feature, Logtail records fine-grained checkpoints by file to the disk of the server on which Logtail is installed. By default, Logtail does not call the sync function to write checkpoints to the disk. However, if buffered data fails to be written to the disk when the server restarts, the checkpoints may be lost. In this case, you can set the enable_checkpoint_sync_write parameter to true to enable the sync write feature. This parameter is available only for Logtail V1.0.20 or later.	"enable_checkpoint_s ync_write": false
enable_env_ref_in_con fig	Boolean	Specifies whether to enable the environment variable replacement feature in Logtail configurations. Default value: false After this feature is enabled, you can use (xxx) as the placeholder for the environment variable xxx when you create a Logtail configuration in the Log Service console. For example, if you set Log Path to $(xxx)/\log s$ and the environment variable to $xxx=root$, the path to the file from which Logtail collects logs is $/root/logs$. If $\{and\}$ are used in your Logtail configuration, you can use $\{s\}$ and $\{s\}$ to escape the characters. This parameter is available only for Logtail V1.0.31 or later.	"enable_env_ref_in_co nfig": false
docker_config_update _interval	int	The minimum interval at which the container path is updated. Default value for versions earlier than Logtail V1.0.32: 10. Default value for Logtail V1.0.32 or later: 3. Unit: seconds. This parameter is used together with the max_docker_config_update_times parameter. If one of the values for the two parameters is reached, the container path is no longer updated.	"docker_config_updat e_interval": 3

max_docker_config_u pdate_times	int	The maximum number of times that the container path can be updated within 3 minutes. Default value for versions earlier than Logtail V1.0.32: 3. Default value for Logtail V1.0.32 or later: 10. By default, if the container path is updated more than three times within a 3-minute period, the container path cannot be updated again until 3 minutes later.	"max_docker_config_u pdate_times": 10
DOCKER_HOST	String	The socket address that is used to communicate with Docker. You must configure the socket address by using environment variables. This parameter is empty by default, which indicates that the default socket address unix:///var/run/docker.sock is used.	DOCKER_HOST=unix:/ //var/run/docker.sock
CONTAINERD_SOCK_P ATH	String	The socket address that is used to communicate with containerd. You must configure the socket address by using environment variables. This parameter is empty by default, which indicates that the default socket address unix:///run/containerd/containerd.sock is used. If a K3s cluster is used, you can change the default socket address to the value provided in the example.	CONTAINERD_SOCK_P ATH=/run/k3s/contain erd/containerd.sock
logreader_max_rotate _queue_size	Int	The maximum length of the queue in which a file is rotated. Default value: 20. If log collection is blocked or delayed, the files from which you want to collect logs are assigned the file handles and wait in the queue. If log collection is delayed and you need to manage the maximum disk usage, you can set this parameter to a smaller value. Marning If the number of delayed files exceeds the value of this parameter, Logtail does not collect logs from new files.	"logreader_max_rotat e_queue_size" : 10
force_release_deleted _file_fd_timeout	Int	The timeout period for the release of a file handle. If you want to release a file handle after a container exits or a file is deleted, you can configure this parameter. Unit: seconds. Default value: -1, which indicates that the feature is disabled. The value 0 indicates that file handles are immediately released. If you want to manage the maximum destruction latency of containerd containers, you can configure this parameter. \bigwedge Warning If log collection is delayed and the latency exceeds the specified threshold, data that is not collected is lost.	"force_release_delete d_file_fd_timeout" : 0
default_max_inotify_w atch_num	Int	The maximum number of directories that are monitored by using inotify. The directories include subdirectories. Default value: 3000.	"default_max_inotify_ watch_num" : 5000

3. Restart Logtail for the new settings to take effect.

/etc/init.d/ilogtaild stop && /etc/init.d/ilogtaild start

After you restart Logtail, you can run the /etc/init.d/ilogtaild status command to check the status of Logtail.

Appendix: Environment variables

The following table describes the mappings between environment variables and the startup parameters of Logtail. For information about the startup parameters of Logtail, see Configure the startup parameters of Logtail.

Table 2. Mappings between environment variables and the startup parameters of Logtail

Parameter	Environment variable	Priority	Supported version
cpu_usage_limit	cpu_usage_limit	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the environment variables take effect.	Logtail V0.16.32 or later
mem_usage_limit	mem_usage_limit	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the environment variables take effect.	Logtail V0.16.32 or later
max_bytes_per_sec	max_bytes_per_sec	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the environment variables take effect.	Logtail V0.16.32 or later
process_thread_count	process_thread_count	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the environment variables take effect.	Logtail V0.16.32 or later

Cloud Defined Storage

User Guide-Log Service

send_request_concurrency	send_request_concurrency	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the environment variables take effect.	Logtail V0.16.32 or later
check_point_filename	check_point_filename or ALIYUN_LOGTAIL_CHECK_POINT_PA TH	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the environment variables take effect.	Logtail V0.16.36 or later
docker_file_cache_path	docker_file_cache_path	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V0.16.54 or later
user_config_file_path	user_config_file_path	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V0.16.56 or later
discard_old_data	discard_old_data	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V0.16.56 or later
working_ip	working_ip or ALIYUN_LOGTAIL_WORKING_IP	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V0.16.56 or later
working_hostname	working_hostname or ALIYUN_LOGTAIL_WORKING_HOSTN AME	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V0.16.56 or later
max_read_buffer_size	max_read_buffer_size	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V0.16.56 or later
oas_connect_timeout	oas_connect_timeout	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V0.16.56 or later
oas_request_timeout	oas_request_timeout	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V0.16.56 or later
data_server_port	data_server_port	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V0.16.56 or later
accept_multi_config	accept_multi_config	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V0.16.56 or later
enable_log_time_auto_adjust	enable_log_time_auto_adjust	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V1.0.19 or later
check_point_dump_interval	check_point_dump_interval	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V1.0.19 or later

enable_checkpoint_sync_write	enable_checkpoint_sync_write	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V1.0.20 or later
docker_config_update_interval	docker_config_update_interval or ALIYUN_LOGTAIL_DOCKER_CONFIG_ UPDATE_INTERVAL	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V1.0.29 or later
max_docker_config_update_times	max_docker_config_update_times or ALIYUN_LOGTAIL_MAX_DOCKER_CO NFIG_UPDATE_TIMES	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V1.0.29 or later
logreader_max_rotate_queue_size	logreader_max_rotate_queue_size	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the configuration file take effect.	Logtail V0.16.54 or later
force_release_deleted_file_fd_timeo ut	force_release_deleted_file_fd_timeo ut	If you use environment variables and the configuration file to modify the startup parameters of Logtail, the modifications from the environment variables take effect.	Logtail V1.21.1 or later

4.3.1.3. Logtail machine group

4.3.1.3.1. Overview

Log Service uses machine groups to manage the servers from which you want to collect logs by using Logtail.

A machine group is a virtual group that contains multiple servers. If you want to use a Logtail configuration file to collect logs from multiple servers, you can add the servers to a machine group. Then, you can apply the Logtail configuration file to the machine group.

To define a machine group, you can use one of the following methods:

- IP address: Add the IP addresses of all servers to a machine group. Each server can be identified by using its unique IP address.
- Custom ID: Use a custom ID to identify the machine group and use the same ID for servers in the machine group.

? Note

Windows and Linux servers cannot be added to the same machine group.

IP address-based machine groups

You can add multiple servers to a machine group by adding their IP addresses to the machine group. Then, you can create a Logtail configuration file for all the servers at the same time.

- If you use ECS instances and have not associated them with hostnames or changed their network types, you can add their private IP addresses to the machine group.
- In other cases, you must add the server IP addresses obtained by Logtail to a machine group. The IP address of each server is recorded in the IP address field of the app_info.json file on the server.

Logtail obtains a server IP address by using the following methods:

- If the IP address of a server is associated with the hostname in the /etc/hosts file of the server, Logtail obtains this IP address.
- If the IP address of a server is not associated with the hostname, Logtail obtains the IP address of the first network interface controller (NIC) on the server.

For more information, see Create an IP address-based machine group.

Custom ID-based machine groups

You can use custom IDs to dynamically define machine groups.

An application system consists of multiple modules. You can scale out each module by adding multiple servers to the module. If you want to collect logs by module, you can create a machine group for each module. Therefore, you must specify a custom ID for each server in each module. For example, a website consists of an HTTP request processing module, a caching module, a logic processing module, and a storage module. The custom IDs of these modules can be http_module, cache_module, logic_module, and store_module.

For more information, see Create a custom identifier-based machine group.

4.3.1.3.2. Create an IP address-based machine group

Log Service allows you to define a machine group by using IP addresses. This topic describes how to create an IP address-based machine group in the Log Service console.

O Note The app_info.json file records the internal information of Logtail. This file includes the server IP addresses obtained by Logtail. If you modify the IP address field of the file, the IP addresses obtained by Logtail remain unchanged.

Prerequisites

- A project and a Logstore are created.
- At least one server is available.
- If you use an Elastic Compute Service (ECS) instance, make sure that the ECS instance belongs to the same Apsara Stack tenant account and region as your Log Service project.
- Logtail is installed on the servers. For more information, see Install Logtail on a Linux server and Install Logtail on a Windows server.

Procedure

- 1. Obtain the IP addresses of servers.
 - The IP addresses that are obtained by Logtail are recorded in the **ip** field of the app_info.json file.
 - To view the app_info.json file, go to the following paths on the servers on which Logtail is installed:
 - Linux: /usr/local/ilogtail/app_info.json
 - 64-bit Windows: C:\Program Files (x86)\Alibaba\Logtail\app_info.json
 - 32-bit Windows: C:\Program Files\Alibaba\Logtail\app_info.json

The following figure shows how to view the IP address of a Linux server.



2. Log on to the Log Service console

- 3. In the Projects section, click the project that you want to manage.
- 4. In the left navigation sidebar, choose **Resources > Machine Groups**.
- 5. Click the picon to the right of Machine Groups and select **Create Machine Group**.

You can also create a machine group in the data import wizard.

6. In the Create Machine Group panel, configure the following parameters and click OK.

Parameter	Description
	Specify the name of the machine group. The name must be 3 to 128 characters in length and can contain only lowercase letters, digits, hyphens (-), and underscores (_). The name must start and end with a lowercase letter or a digit.
Name	① Important After the machine group is created, you cannot change the name of the machine group. Proceed with caution.
Identifier	Select IP Addresses.
Торіс	Specify the topic of the machine group. The topic is used to differentiate the logs that are generated by different servers. For more information, see Log topics.
	Enter the IP addresses of servers that you obtain inStep 1.
IP Addresses	 Note If you want to add multiple servers to a machine group, separate the IP addresses with line feeds. Do not add Windows and Linux servers to the same machine group.

7. View the status of the machine group.

- i. In the Machine Groups list, click the machine group that you create.
- ii. On the Machine Group Settings page, view the server details and machine group status.
- If a value in the **Heartbeat** column is **OK**, the server is connected to Log Service. If the value is **FAIL**, the connection failed. In this case, click **Automatic Retry**.

Server Group Status	
IP V Enter the IP address	Q Total:1
IP	Heartbeat 7
102.100.0.105	ОК

Result

You can view the machine group that is created in the Machine Groups list.

Machine Groups	Endpoint List	Create Machine Group
Searching by group name Search		
Group Name		Action
test	Modify Ma	achine Status Config Delete

4.3.1.3.3. Create a custom identifier-based machine group

Log Service allows you to define a machine group by using a custom identifier. This topic describes how to create a custom identifier-based machine group.

Prerequisites

- A project and a Logstore are created. For more information, see Create a project and Create a Logstore.
- At least one server is available.
 - If you use an Elastic Compute Service (ECS) instance, make sure that the ECS instance belongs to the same Apsara Stack tenant account and region as your Log Service project.
 - Logtail is installed on the servers. For more information, see Install Logtail on an ECS instance.

Background information

Custom identifier-based machine groups provide benefits in the following scenarios:

- If your servers reside in multiple custom network environments such as virtual private clouds (VPCs), the IP addresses of some servers may conflict. In this case, Log Service cannot manage Logtail. You can create a custom identifier-based machine group to prevent this issue.
- If you want to achieve auto scaling for a machine group, you need to only configure the same custom identifier for new servers that you want to add to the machine group. Log Service identifies the custom identifier and adds the servers that have the same custom identifier to the same machine group.

Procedure

- 1. Create a file named user_defined_id in a specified directory.
 - Linux server: Create the file in the /etc/ilogtail/user_defined_id directory.
 - Windows server: Create the file in the C:\LogtailData\user_defined_id directory.
- 2. Configure a custom identifier for your servers.

? Note

- Windows and Linux servers cannot be added to the same machine group. Do not configure the same custom identifier for Linux and Windows servers.
- You can configure one or more custom identifiers for a single server and separate custom identifiers with line feeds.
- On a Linux server, if the /etc/ilogtail/ directory or the /etc/ilogtail/user_defined_id file does not exist, you must create the directory and the file. On a Windows server, if the C:\LogtailData directory or the C:\LogtailData\user_defined_id file does not exist, you must also create the directory and the file.

Linux server

Configure a custom identifier in the /etc/ilogtail/user_defined_id file. For example, if you want to configure a custom identifier as userdefined , run the following command to edit the file. Then, enter userdefined in the file and save the file.

vim /etc/ilogtail/user_defined_id

• Windows server

Configure a custom identifier in the C:\LogtailData\user_defined_id file. For example, if you want to configure a custom identifier as userdefined_windows , enter userdefined_windows in the C:\LogtailData\user_defined_id file and save the file.

3. Create a machine group.

- i. Log on to the Log Service console
- ii. In the Projects section, click the project that you want to manage.
- iii. In the left-side navigation pane, choose **Resources > Machine Groups**.
- iv. Click the $_{\mbox{\tiny MR}}$ icon to the right of Machine Groups and select Create Machine Group.

v. Configure the machine group

Parameter	Description
Name	Specify the name of the machine group. The name must be 2 to 128 characters in length, and can contain only lowercase letters, digits, hyphens (-), and underscores (_). The name must start and end with a lowercase letter or a digit.
Identifier	Select Custom ID .
Торіс	Specify the topic of the machine group. The topic is used to differentiate the logs that are generated by different servers. For more information, see Log topics.
Custom Identifier	Enter the custom identifier that is configured in step of "Configure a custom identifier".

vi. Click OK

? Note

- To scale out servers, you need to only configure the custom identifier for new servers.
- 4. In the Machine Groups list, click the machine group and check the status of the machine group.

On the machine group configuration page, you can view the server list and the heartbeat status of the servers in the Server Group Status section.

Server Group Status							
Heartbeat V Enter the IP address		Q	Total:1				
IP	Heartbeat 7						
15	ОК						

• The Server Group Status section displays the IP addresses of the servers whose custom identifier is the same.

For example, a custom identifier-based machine group is created, the custom identifier is userdefined, and the IP addresses in the Server Group Status section are 10.10.10.10, 10.10.10.11, and 10.10.10.12. This indicates that you configured the same custom identifier for the servers in the machine group. If you want to add another server to the machine group and the IP address of the server is 10.10.10.13, configure the custom identifier as userdefined for the server. Then, you can view the IP address of the server that you added in the Server Group Status section. Log Service collects logs from the server based on the Logtail configuration of the machine group.

 The heartbeat status indicates whether a server is connected to Log Service. If FAIL is displayed in the heartbeat column, the connection failed. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail client?

Disable a custom identifier

If you want to change the identifier of a server from a custom identifier to the server IP address, delete the user_defined_id file. The modification takes effect within 1 minute.

• Linux

rm -f /etc/ilogtail/user_defined_id

Windows

del C:\LogtailData\user_defined_id

Effective time

By default, after you add, delete, or modify the user_defined_id file, the new configuration takes effect within 1 minute.

• Linux

```
/etc/init.d/ilogtaild stop
/etc/init.d/ilogtaild start
```

- Windows
- i. Choose Start Menu > Control Panel > Administrative Tools > Services.
- ii. In the Services dialog box, select the service that you want to manage.
 - For Logtail V0.x.x.x, select LogtailWorker.
 - For Logtail V1.0.0.0 or later, select LogtailDaemon.
- iii. Right-click the service and select **Restart** for the configuration to take effect.

4.3.1.3.4. View server groups

This topic describes how to view the server groups of a project on the Server Groups page in the Log Service console.

Procedure

- 1. Log on to the Log Service console
- 2. Find the target project in the project list and click the project name.
- 3. In the left-side navigation pane of the page that appears, click the Server Groups icon to display the list of server groups.

You can view all server groups of the project.

< sls-test		Switch	6	⊘ doctes	st	×	10-17	-5-24 X	
B Recent Visits		Machine Groups	88	М	Machine Group Settings (10-17-5-24)				
		Search machine group name Q							
Log Storage		• 10-17-5-24				Mac	hine Group	Details	
Time Series Stora	age						* Name:	10-17-5-24	
Resources	~						Identifier:	IP Addresse	es V
🕒 Dashboard							Topic:		
Jobs	~					* IF	Addresses		
Alerts									
0ther	~						Note:	1. Currently, or the project are	nly ECS instances in the same region as e supported.
								2. Enter the int Separate multi	ternal IP addresses of the ECS instances. inle IP addresses with line breaks
								3. Windows an same machine	ind Linux instances are not allowed in the group.
						Mac	hine Group	Status	

4.3.1.3.5. Modify a server group

This topic describes how to modify a server group in the Log Service console. After you create a server group, you can modify the parameters of the server group.

Procedure

- 1. Log on to the Log Service console
- 2. Find the target project in the project list and click the project name.
- 3. In the left-side navigation pane of the page that appears, click the Server Groups icon to display the list of server groups.
- 4. Click the name of the server group to be modified. On the Server Group Settings page, click Modify.

```
⑦ Note
The name of the server group cannot be modified.
```

5. Modify the parameters of the server group, and then click Save.

4.3.1.3.6. View the status of a server group

This topic describes how to view the status of a server group in the Log Service console. You can view the heartbeat information of Logtail to check whether Logtail is installed on the servers in a server group.

Procedure

- 1. Log on to the Log Service console
- 2. Find the target project in the project list and click the project name.
- 3. In the left-side navigation pane of the page that appears, click the Server Groups icon to display the list of server groups.
- 4. Click the name of the server group. On the Server Group Settings page, check the server group status.
- If the heartbeat is OK, Logtail is installed on the servers in the server group and Logtail is connected to Log Service.
- If the heartbeat status is FAIL, Logtail fails to connect to Log Service. If the FAIL state persists, perform troubleshooting based on the instructions
 provided in What can I do if Log Service does not receive heartbeats from a Logtail client?

4.3.1.3.7. Delete a machine group

This topic describes how to delete a machine group in the Log Service console. You can delete a machine group if you no longer need to collect logs from the machine group.

Procedure

- 1. Log on to the Log Service console
- 2. In the Projects section, click the project in which you want to delete a machine group.
- 3. In the left-side navigation pane, click the icon. The Machine Groups list is displayed.

4. In the Machine Groups list, find the machine group that you want to delete, click the icon next to the machine group, and then select **Delete**.

5. In the message that appears, click **OK**.



4.3.1.3.8. Manage a Logtail configuration

This topic describes how to create, view, modify, and delete a Logtail configuration in the Log Service console.

View a list of Logtail configurations

1. Log on to the Log Service console

- 2. In the Projects section, click the project in which you want to view Logtail configurations.
- 3. Choose Log Storage > Logstores. On the Logstores tab, Click the > icon of the Logstore in which you want to view Logtail configurations. Then, choose Data Import > Logtail Configurations.
- 4. Click the Logtail configuration that you want to view.
- 5. On the Logtail Config page, view the details of the Logtail configuration.

Create a Logtail configuration

You can create a Logtail configuration in the Log Service console. For more information, see Configure text log collection.

Modify a Logtail configuration

Click the name of the Logtail configuration that you want to modify. On the Logtail Config page, click Modify.

You can also change the log collection mode of the Logtail configuration, and then apply the Logtail configuration to the related machine group again. The procedure to modify a Logtail configuration is the same as the procedure to create a Logtail configuration.

Delete a Logtail configuration

In the Logtail Configurations list, find the Logtail configuration that you want to delete, click the 📓 icon next to the Logtail configuration, and then select Delete.

🔥 Warning

After you delete a Logtail configuration, the Logtail configuration is disassociated from the related machine group. Logtail no longer collects the logs that are specified by the Logtail configuration. Proceed with caution.

4.3.1.4. Collect text logs

4.3.1.4.1. Configure text log collection

This topic describes the configuration process and collection modes when you use Logtail to collect text logs from servers.

Prerequisites

A project and a Logstore are created. For more information, see Create a project and Create a Logstore.

Configuration process

Log Service provides a configuration wizard that you can use to configure log collection.

1	2	3	4	5	6 -
Specify Logstore	Create Server Group	Server Group Settings	Logtail Config	Configure Query and Analysis	End

Collection modes

Logtail supports various collection modes, such as simple mode, full regex mode, delimiter mode, JSON mode, NGINX configuration mode, IIS configuration mode, and Apache configuration mode.

- Collect logs in simple mode
- Collect logs in full regex mode
- Collect logs in delimiter mode
- Collect logs in JSON mode
- Collect logs in NGINX mode
- Collect logs in IIS mode
- Collect logs in Apache mode

Procedure

- 1. Log on to the Log Service console
- 2. In the Import Data section, select a data source.

Select a data source based on your business requirements. Log Service supports the following data sources of text logs: RegEx - Text Log, Single Line - Text Log, Multi-Line - Text Log, Delimiter Mode - Text Log, JSON - Text Log, Nginx - Text Log, IIS - Text Log, and Apache - Text Log.

Select a destination project and Logstore, and then click Next.
 You can also click Create Now to create a project and a Logstore.

If you enter the configuration procedure by clicking the plus sign (+) next to **Data Import** under a Logstore on the Logstores tab, the system skips this step.

4. Create a machine group and click $\ensuremath{\textbf{Next}}.$

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. For more information, see Install Logtail on a Linux server or Install Logtail on a Windows server.

After you install Logtail, click Complete Installation to create a machine group. If a machine group is created, click Use Existing Machine Groups to select the machine group.

5. Select a machine group, move the machine group from Source Machine Groups to Applied Server Groups, and then click Next.

Source Server Groups			Applied Server Groups	
Search by server group name	Q		Search by server group name	Q
	d			
		>		
		<		
1 Items			0 Items	
Thomas				

() Important

If you want to apply a machine group immediately after it is created, the heartbeat status of the machine group may be **FAIL**. This issue occurs because no servers in the machine group are connected to Log Service. In this case, you can click **Automatic Retry**.

6. Create a Logtail configuration and click Next.

Logtail parameters vary based on collection modes. For more information, see the related parameters for specific collection methods in Collection modes.

Parameter	Description
Config Name	The name of the Logtail configuration. The name must be unique in a project. After the Logtail configuration is created, you cannot change the name of the Logtail configuration. You can also click Import Other Configuration to import a Logtail configuration from another project.
Log Path	 The directory and name of the log file. The specified log file name can be a complete file name or a file name that contains wildcards. Log Service scans all levels of the specified directory to match log files. Examples: If you specify /apsara/nuwa//*.log, Log Service matches the files whose name is suffixed by.log in the /apsara/nuwa directory and its recursive subdirectories. If you specify //ar/logs/app_*/*.log, Log Service matches the files that meet the following conditions: The file name contains. Jog. The file is stored in a subdirectory of the/var/logs directory or in a recursive subdirectory of the subdirectory matches the app_* pattern. Note By default, each log file can be collected by using only one Logtail configuration. To use multiple Logtail configurations to collect one log file, we recommend that you create a symbolic link that points to the directory. Where the file is located. For example, you want to collect two copies of the log.log file from the /home/log/nginx/log/log.log directory. You can run the following command to create a symbolic link that points to the directory. When you configure the Logtail configuration. In -s /home/log/nginx/log /home/log/nginx/link_log You can use only asterisks (*) and question marks (?) as wildcards in the log path.
Blacklist	If you turn on Blacklist , you can configure a blacklist to skip the specified directories or files when Logtail collects logs. You can use exact match or wildcard match to specify directories and files. Examples: • If you select Filter by Directory from the Filter Type drop-down list and enter/tmp/mydir in the Content column, all files in the directory are skipped. • If you select Filter by File from the Filter Type drop-down list and enter/tmp/mydir/file in the Content column, only the specified file is skipped.

Docker File	If you want to collect logs from Docker containers, you can turn on Docker File and specify the paths and tags of the containers. Logtail monitors the containers to check whether the containers are created or destroyed. Then, Logtail filters the logs of the containers based on tags and collects the filtered logs. For more information about container text logs, see Install the Logtail component.		
Mode	The default mode is Simple Mode - Multi-line . You can change the mode.		
Log Sample	Enter a sample log that is retrieved from a log source in an actual scenario. Then, Log Service can automatically generate a regular expression to match the start part in the first line of the log. Example:		
	<pre>[2020-10-01T10:30:01,000] [INFO] java.lang.Exception: exception happened at TestPrintStackTrace.f(TestPrintStackTrace.java:3) at TestPrintStackTrace.g(TestPrintStackTrace.java:7) at TestPrintStackTrace.main(TestPrintStackTrace.java:16)</pre>		
	If you collect single-line text logs in simple mode, you do not need to set this parameter.		
	The regular expression that Logtail uses to match the start part in the first line of a log. The unmatched lines are collected as part of a log. You can specify a regular expression to match the start part in the first line of a log. You can also use the regular expression that is automatically generated by Log Service.		
	Automatically generate a regular expression to match the start part in the first line of a log.		
Regex to Match First Line	After you enter a sample log, click Auto Generate . Log Service automatically generates a regular expression to match the start part in the first line of the log.		
	 Specify a regular expression to match the start part in the first line of a log. 		
	After you enter a sample log, click Manual and specify a regular expression to match the start part in the first line of the log. Then, click Validate to check whether the regular expression is valid.		
	If you collect single-line text logs in simple mode, you do not need to set this parameter.		
Drop Failed to Parse Logs	Specifies whether to drop logs that fail to be parsed.		
	 If you turn on Drop Failed to Parse Logs, logs that fail to be parsed are not uploaded to Log Service. 		
	• If you turn off Drop Failed to Parse Logs, raw logs are uploaded to Log Service if the logs fail to be parsed.		
Maximum Directory Monitoring Depth	The maximum depth at which the log directory is monitored. Valid values: 0 to 1000. The value 0 indicates that only the specified log file directory is monitored.		

7. Optional:Specify Advanced Options and click Next.

Set the parameters in the Advanced Options section based on your business requirements. We recommend that you do not modify the settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specifies whether to enable plug-in processing. If you turn on this switch, you can use the plug-in of Logtail to process text logs.
	② Note If you turn on Enable Plug-in Processing , specific parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.
Upload Raw Log	If you turn on Upload Raw Log , each raw log is uploaded to Log Service as a value of the_ raw_ field together with the parsed log.
Topic Generation Mode	 The topic generation mode. Null - Do not generate topic: This mode is selected by default. In this mode, the topic field is set to an empty string. You do not need to enter a topic to query logs. Machine Group Topic Attributes: This mode is used to differentiate logs that are generated by different servers. File Path RegEx: In this mode, you must specify a regular expression in theCustom RegEx field. The part of a log path that matches the regular expression is used as the topic. This mode is used to differentiate logs that are generated by different expression is used as the topic. This mode is used to differentiate logs that are generated by different users or instances.
Custom RegEx	If you set the Topic Generation Mode parameter to File Path RegEx, you must enter a custom regular expression.
Log File Encoding	The encoding format of log files. Valid values: • utf8: UTF-8 encoding format • gbk: GBK encoding format
Timezone	 The time zone where logs are collected. Valid values: System Timezone: This option is selected by default. It indicates that the time zone where logs are collected is the same as the time zone to which the server belongs. Custom: If you select this value, you must select a time zone.

Timeout	 The timeout period of log files. If a log file is not updated within the specified period, Logtail reckons the file to be timed out. Valid values: Never: All log files are continuously monitored and never time out. 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select 30 Minute Timeout, you must specify the Maximum Timeout Directory Depth parameter. Valid values: 1 to 3.
Filter Configuration	 Only logs that meet all filter conditions are collected. Examples: Collect logs that meet specified conditions: If you setKey to level and Regex to WARNING ERROR, only WARNING-level and ERROR-level logs are collected. Filter out logs that do not meet specified conditions. If you set Key to level and Regex to ^(?!.*(INFO DEBUG)).*, INFO-level or DEBUG-level logs are not collected. If you set Key to url and Regex to .*^(?!.*(INFO DEBUG)).*, logs whose URLs contain healthcheck are not collected. For example, the log is not collected if the value of the Key field is url and the value of the Value field is /inner/healthcheck/jiankong.html.

8. Configure indexes in the Configure Query and Analysis step. Click Next.

By default, Log Service enables Full Text Index to query and analyze logs. You can manually or automatically configure Field Search. For more information, see Configure indexes.

? Note

To query and analyze logs, you must enable Full Text Index or Field Search. If you enable both Full Text Index and Field Search, the settings
of Field Search prevail.

• If the data type of the index is long or double, the Case-Sensitive switch and the Delimiter field are unavailable.

After you complete the settings, you can start to collect logs.

4.3.1.4.2. Collect logs in simple mode

When you collect logs in simple mode, the logs are not parsed. Each log is collected and uploaded to Log Service as a whole. This simplifies the process of log collection. This topic describes how to create a Logtail configuration in simple mode in the Log Service console to collect logs.

Prerequisites

- A project and a Logstore are created. For more information, see Create a project and Create a Logstore.
- Ports 80 and 443 are enabled for the server from which you want to collect logs.

Background information

The simple mode supports the following types of text logs:

Single-line text log

Each log line is collected as a log. Two logs in a log file are separated by a line feed. In single-line mode, you must specify the directories and names of log files. Then, Logtail collects logs by line from the specified files.

Multi-line text log

Multiple log lines are collected as a log by default. In multi-line mode, you must specify the directories and names of log files. In addition, you must enter a sample log and configure a regular expression to match the start part in the first line of a log. Logtail uses the regular expression to match the start part in the first line of a log and reckons unmatched lines as part of the log.

? Note

If you collect logs in simple mode, the timestamp of a log indicates the system time of the server when the log is collected.

Procedure

- 1. Log on to the Simple Log Service console
- 2. Select a data source.
- Select Single Line Text Log
- 3. Select the project and Logstore. Then, click Next.
 - You can also click **Create Now** to create a project and a Logstore.

If you click the plus sign (+) next to **Data Import** below your Logstore to enter the configuration process, the system automatically skips this step.

Create a machine group and click **Next**.

Before you create a machine group, make sure that Logtail is installed.

Follow the on-screen instructions to install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows.

After you install Logtail, click **Complete Installation** to create a machine group. If you created a machine group, click **Use Existing Machine Groups**.

 Select the machine group in the Source Server Groups section and move the machine group to the Applied Server Groups section. Then, click Next.

() Important

If you apply a machine group immediately after you create the machine group, the heartbeat status of the machine group may be **FAIL**. This issue occurs because the machine group is not connected to Simple Log Service. To resolve the issue, you can click **Automatic Retry**. If the status is still **FAIL**, see What do I do if a Logtail machine group has no heartbeats?

Source Server Groups			Applied Server Groups	
Search by server group name	Q.		Search by server group name	Q
		>		
		<		
1 Items			0 Items	

6. Create a Logtail configuration.

The following table describes the Logtail parameters.

Parameter	Description
Config Name	The name must be 3 to 128 characters in length, and can contain only lowercase letters, digits, hyphens (-), and underscores (_). The name must start and end with a lowercase letter or a digit.
Log Path	 The directory and name of the log file. The specified log file name can be a complete file name or a file name that contains wildcards. Recursive directory matching is used in the log file search. If this matching method is applied, all files that match the specified file name in the specified directory and its sub-directories are monitored. If you specify /apsara/nuwa/ ** /*.log , Log Service matches the files whose name is suffixed by .log in the /apsara/nuwa directory and its recursive subdirectories. If you specify /var/logs/app_* /*.log* , Log Service matches the files that meet the following conditions: The file name contains .log . The file is stored in a subdirectory of the /var/logs directory or in a recursive subdirectory of the subdirectory. The name of the subdirectory matches the app_* pattern. Note By default, each log file can be collected by using only one Logtail configuration. You can use only asterisks (*) and question marks ?) as wildcards in the log path.
Docker File	If you want to collect logs from Docker containers, you can specify the paths and tags of the containers. Logtail monitors the containers to check whether the containers are created or destroyed. Then, Logtail filters the logs of the containers based on tags and collects the filtered logs.
Mode	If you have specified Single Line - Text Log for the data source, the default mode is Simple Mode . You can change the mode.
Maximum Directory Monitoring Depth	The maximum number of directory layers that can be recursively monitored when logs are collected from the data source. Valid values: 0 to 1000. The value 0 indicates that only the specified log file directory is monitored.

7. Optional. Configure parameters in the Advanced Options section. After the settings are complete, click Next.

You can configure advanced settings based on your business requirements. We recommend that you retain the default settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description	
Enable Plug-in Processing	Specify whether to enable plug-in processing. If you turn on this switch, you can use plug-ins to process text logs. Note If you turn on Enable Plug-in Processing, the parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.	

User Guide-Log Service

Upload Raw Log	If you turn on Upload Raw Log , each raw log is uploaded to Simple Log Service as the value of the_ raw _ field together with the log parsed from the raw log.
Topic Generation Mode	 Select the topic generation mode. Null - Do not generate topic: In this mode, the topic field is set to an empty string. When you query logs, you do not need to specify a topic. This is the default value. Machine Group Topic Attributes: In this mode, topics are configured at the machine group level. If you want to distinguish the logs that are generated by different servers, select this mode. File Path RegEx: In this mode, you must specify a regular expression in theCustom RegEx field. The part of a log path that matches the regular expression is used as the topic. If you want to distinguish the logs that are generated by different servers.
Custom RegEx	Specify a custom regular expression. If you select File Path RegEx for Topic Generation Mode, you must specify a custom regular expression.
Log File Encoding	Select the encoding format of log files. Valid values: utf8 and gbk. • utf8: UTF-8 encoding is used. • gbk: GBK encoding is used.
Timezone	 Select the time zone for the time of logs that are collected. Valid values: System Timezone: If you select this value, the time zone of the server is used. Custom: If you select this value, you must select a time zone based on your business requirements.
Timeout	 If a log file is not updated within the specified period, Logtail considers the log file to be timed out. Valid values: Never: All log files are continuously monitored, and log files never time out. 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select 30 Minute Timeout, you must configure Maximum Timeout Directory Depth. Valid values: 1 to 3.
Filter Configuration	 Specify the conditions to filter logs. Only logs that exactly match the specified filter conditions are collected. Examples: Collect the logs that match the specified filter conditions. If you setKey to level and set Regex to WARNING/ERROR, only logs in which the value of level is WARNING or ERROR are collected. Filter out the logs that do not match the specified filter conditions. If you set Key to level and set Regex to ^(?!.*(INFO]DEBUG)).*, logs in which the value of level is INFO or DEBUG are filtered out. If you set Key to url and set Regex to .*^(?!.*(healthcheck)).*, logs in which the value of url contains healthcheck are filtered out.

8. Configure indexes for query and analysis. After you complete the settings, click Next.

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on the collected logs in manual or automatic mode. For more information, see Enable the index feature and configure indexes for a Logstore.

? Note

- If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.
- If you configure an index for a field of the Long or Double type, you cannot configure Case Sensitive or Delimiter for the field.

After you complete the settings, you can start to collect logs in full regex mode.

4.3.1.4.3. Collect logs in full regex mode

You can use the full regex mode to extract custom fields from logs. This topic describes how to create a Logtail configuration in full regex mode in the Log Service console to collect logs.

Background information

If you want to collect multi-line logs and extract fields from the logs, we recommend that you use regular expressions. Log Service can generate a regular expression based on a sample log that you specify in the Import Data wizard. However, you must modify a regular expression before it can match fields in the sample log as expected. For more information, see How do I debug a regular expression?

Procedure

- 1. Log on to the Simple Log Service console
- 2. Select a data source.
- Select RegEx Text Log
- 3. Select the project and Logstore. Then, click Next.

You can also click Create Now to create a project and a Logstore.

If you click the plus sign (+) next to **Data Import** below your Logstore to enter the configuration process, the system automatically skips this step. 4. Create a machine group and click **Next**.

Before you create a machine group, make sure that Logtail is installed.

Follow the on-screen instructions to install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows.

After you install Logtail, click **Complete Installation** to create a machine group. If you created a machine group, click **Use Existing Machine Groups**.

5. Select the machine group in the Source Server Groups section and move the machine group to the Applied Server Groups section. Then, click Next.

! Important

If you apply a machine group immediately after you create the machine group, the heartbeat status of the machine group may be FAIL. This issue occurs because the machine group is not connected to Simple Log Service. To resolve the issue, you can click **Automatic Retry**. If the status is still **FAIL**, see What do I do if a Logtail machine group has no heartbeats?

ource Server Groups		Applied Server Groups	
Search by server group name	Q	Search by server group name	Q
1 Items		0 Items	

6. Create a Logtail configuration.

The following table describes the Logtail parameters.

Parameter	Description		
Config Name	The name of the Logtail configuration. The name must be unique in a project. After the Logtail configuration is created, you cannot change the name of the Logtail configuration. You can also click Import Other Configuration to import a Logtail configuration from another project.		
Log Path	 The directory and name of the log file. The specified log file name can be a complete file name or a file name that contains wildcards. Log Services scans all levels of the specified directory to match log files. Examples: If you specify /apsara/nuwa/**/*.log, Log Service matches the files whose name is suffixed by.log in the /apsara/nuwa directory and its recursive subdirectories. If you specify /var/logs/app_*/*.log, Log Service matches the files that meet the following conditions: The file name contains .log. The file is stored in a subdirectory of the/var/logs directory or in a recursive subdirectory of the subdirectory. The name of the subdirectory matches the app_* pattern. Note By default, each log file can be collected by using only one Logtail configuration. To use multiple Logtail configurations to collect one log file, we recommend that you create a symbolic link that points to the directory where the file is located. For example, you want to collect two copies of the log.log file from the /home/log/nginx/log/log.log directory. Vou can run the following command to create a symbolic link that points, use the original path in one Logtail configuration and use the symbolic link in the other Logtail configuration. In -s /home/log/nginx/log /home/log/nginx/link_log You can use only asterisks (*) and question marks (?) as wildcards in the log path. 		
Docker File	If you collect logs from Docker containers, you can configure the paths and tags of the containers. Logtail monitors the containers when they are created and destroyed, filters the logs of the containers by tag, and collects the filtered logs.		
Mode	If you have specified RegEx - Text Log for the data source, the default mode is Full Regex Mode . You can change the mode.		
Singleline	The singleline mode is enabled by default. In this mode, logs are separated by line. To collect multi-line logs, such as Java program logs, you must disable the Singleline mode and configure Regex to Match First Line .		
Log Sample	Enter a sample log that is retrieved from a log source in an actual scenario. Then, Log Service can automatically generate a regular expression.		

Regex to Match First Line	You can click Auto Generate or Manual . After you enter a sample log and click Auto Generate , Log Service automatically generates a regular expression. If no regular expression is generated, you can switch to the manual mode and enter a regular expression for verification.
Extract Field	To analyze and process specific fields in logs, you can turn on Extract Field . Then, the specified fields are converted to key-value pairs and sent to Log Service. You must specify a regular expression to parse the log content.
Regular Expression	 If you turn on Extract Field, you must specify this parameter. Automatically generate a regular expression. You can select the fields to be extracted from the sample log and then click Generate Regular Expression. Specify a regular expression. You can also enter a regular expression. ClickManual to switch to the manual mode. After you enter a regular expression, click Validate to check whether Log Service can parse the log content by using the regular expression. For more information, see How do I debug a regular expression?
Extracted Content	If you turn on Extract Field, you must specify this parameter. After a regular expression is automatically generated or manually specified, you must specify the key name for each extracted field.
Use System Time	If you turn on Extract Field, you must specify this parameter. If you turn off Use System Time, you must specify a field as the time field and name the field time. After you specify the time field, click Auto Generate in the Time Conversion Format field to parse the time. For more information, see Time formats.
Drop Failed to Parse Logs	 Specifies whether to upload logs to Log Service if the logs fail to be parsed. If you turn on this switch, logs that fail to be parsed are not uploaded to Log Service. If you turn off this switch, raw logs are uploaded to Log Service if the logs fail to be parsed.
Maximum Directory Monitoring Depth	The maximum number of directory levels that can be recursively monitored when Log Service collects logs from the data source. Valid values: 0 to 1000. The value 0 indicates that only the specified log file directory is monitored.

7. Optional. Configure parameters in the Advanced Options section. After the settings are complete, click Next.

You can configure advanced settings based on your business requirements. We recommend that you retain the default settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specify whether to enable plug-in processing. If you turn on this switch, you can use plug-ins to process text logs.
	O Note If you turn on Enable Plug-in Processing, the parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.
Upload Raw Log	If you turn on Upload Raw Log , each raw log is uploaded to Simple Log Service as the value of the raw field together with the log parsed from the raw log.
	Select the topic generation mode.
Topic Generation Mode	• Null - Do not generate topic: In this mode, the topic field is set to an empty string. When you query logs, you do not need to specify a topic. This is the default value.
	 Machine Group Topic Attributes: In this mode, topics are configured at the machine group level. If you want to distinguish the logs that are generated by different servers, select this mode.
	• File Path RegEx: In this mode, you must specify a regular expression in the Custom RegEx field. The part of a log path that matches the regular expression is used as the topic. If you want to distinguish the logs that are generated by different users or instances, select this mode.
Custom RegEx	Specify a custom regular expression. If you select File Path RegEx for Topic Generation Mode, you must specify a custom regular expression.
Log File Encoding	Select the encoding format of log files. Valid values: utf8 and gbk.
	 utf8: UTF-8 encoding is used. gbk: GBK encoding is used.
	Select the time zone for the time of logs that are collected. Valid values:
Timezone	• System Timezone: If you select this value, the time zone of the server is used.
	 Custom: If you select this value, you must select a time zone based on your business requirements.

Timeout	 If a log file is not updated within the specified period, Logtail considers the log file to be timed out. Valid values: Never: All log files are continuously monitored, and log files never time out. 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select 30 Minute Timeout, you must configure Maximum Timeout Directory Depth. Valid values: 1 to 3.
Filter Configuration	 Specify the conditions to filter logs. Only logs that exactly match the specified filter conditions are collected. Examples: Collect the logs that match the specified filter conditions. If you setKey to level and set Regex to WARNING ERROR, only logs in which the value of level is WARNING or ERROR are collected. Filter out the logs that do not match the specified filter conditions. If you set Key to level and set Regex to ^(?!.*(INFO DEBUG)).*, logs in which the value of level is INFO or DEBUG are filtered out. If you set Key to url and set Regex to .*^(?!.*(INFO DEBUG)).*, logs in which the value of url contains healthcheck are filtered out. For example, if a log contains a field whose key is url and value is /inner/healthcheck/jiankong.html, the log is filtered out.

8. Configure indexes for query and analysis. After you complete the settings, click Next.

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on the collected logs in manual or automatic mode. For more information, see Enable the index feature and configure indexes for a Logstore.

? Note

- If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.
- If you configure an index for a field of the Long or Double type, you cannot configure Case Sensitive or Delimiter for the field.

After you complete the settings, you can start to collect logs in full regex mode.

4.3.1.4.4. Collect logs in delimiter mode

Log Service allows you to collect logs in delimiter mode. After logs are collected, you can transform and ship the logs, and perform multidimensional log analysis. This topic describes how to create a Logtail configuration in delimiter mode in the Log Service console to collect logs.

Prerequisites

- A project and a Logstore are created. For more information, see Create a project and Create a Logstore.
- Ports 80 and 443 are enabled for the server from which you want to collect logs.

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Import Data section, select Delimiter Mode Text Log.
- Select the project and Logstore. Then, click Next.
 You can also click Create Now to create a project and a Logstore.

If you click the plus sign (+) next to **Data Import** below your Logstore to enter the configuration process, the system automatically skips this step. 4. Create a machine group and click **Next**.

Before you create a machine group, make sure that Logtail is installed.

Follow the on-screen instructions to install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows.

After you install Logtail, click **Complete Installation** to create a machine group. If you created a machine group, click **Use Existing Machine Groups**.

5. Select the machine group in the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

() Important

If you apply a machine group immediately after you create the machine group, the heartbeat status of the machine group may be **FAIL**. This issue occurs because the machine group is not connected to Simple Log Service. To resolve the issue, you can click **Automatic Retry**. If the status is still **FAIL**, see What do I do if a Logtail machine group has no heartbeats?

Source Server Groups			Applied Server Groups	
Search by server group name	Q		Search by server group name	Q
<mark>/</mark>	d			
		>		
		<		
1 Items			0 Items	

6. Create a Logtail configuration and click Next.

The following table describes the Logtail parameters.

Parameter	Description
Config Name	The name of the Logtail configuration. The name must be 3 to 63 characters in length, and can contain only lowercase letters, digits, hyphens (-), and underscores (_). The name must start and end with a lowercase letter or a digit.
Log Path	 The directory and name of the log file. You can specify an exact directory and an exact name. You can also use wildcard characters to specify the directory for the log files that match specified conditions. Examples: If you specify /apsara/nuwa/**/*.log, Simple Log Service collects logs from the log files whose names are suffixed by .log in the /apsara/nuwa directory and the recursive subdirectories of the directory. If you specify /var/logs/app_*/*.log, Simple Log Service collects logs from the log files that meet the following conditions: The file name contains. log. The file is stored in a subdirectory under the/var/logs directory or in a recursive subdirectory of the subdirectory. The name of the subdirectory matches the app_* pattern. Nete By default, you can use only one Logtail configuration to collect logs from a log file. you must create a symbolic link for the directory in which the log file is stored in a subdirectory of the sub optical configurations to collect logs from a log file, you want to use two Logtail configurations to collect logs from a log file, you want to use two Logtail configurations to collect logs from a log file, you want to use two Logtail configuration and specify the symbolic link for the directory in which the log file is stored. For example, if you want to use two Logtail configuration and specify the symbolic link in the other Logtail configuration. In -s /home/log/nginx/log /home/log/nginx/link_log When you configure this parameter, you can use only asterisks (*) or question marks (?) as wildcard characters. You can use a question mark (?) to match a single character.
Docker File	If you want to collect logs from Docker containers, you can specify the paths and tags of the containers. Logtail monitors the containers to check whether the containers are created or destroyed. Then, Logtail filters the logs of the containers based on tags and collects the filtered logs.
Mode	The default mode is Delimiter Mode . You can change the mode.
Log Sample	Enter a sample log that is retrieved from a log source in an actual scenario. Example: 127.0.0.1 # - # 13/Apr/2020:09:44:41 +0800 # GET /1 HTTP/1.1 # 0.000 # 74 # 404 # 3650 # - # curl/7.29.0 ⑦ Note The delimiter mode applies only to single-line logs. If you want to collect multi-line logs, we recommend that you select Simple Mode - Multi-line or Full Regex Mode.

Delimiter	Select a delimiter based on the log format. For example, if you select Vertical Line, a vertical bar () is used as the delimiter. For more information, see Appendix: delimiters and sample logs. Image: The second seco
Quote	If a log field contains delimiters, you must specify a pair of quotes to enclose the field. Log Service parses the content that is enclosed in a pair of quotes into a complete field. Select a quote based on the log format. ⑦ Note If you set the Quote parameter to Hidden Characters, you must enter a character in the following format: 0xHexadecimal ASCII code of the non-printable character For example, if you want to use the non-printable character whose hexadecimal ASCII code is 01, you must enter 0x01 .
Extracted Content	Log Service extracts the log content based on the sample log and delimiter that you specify. The extracted log content is delimited into values. You must specify a key for each value.
Incomplete Entry Upload	 Specifies whether to upload a log whose number of parsed fields is less than the number of the specified keys. If you turn on this switch, the log is uploaded. If you turn off this switch, the log is dropped. For example, if you specify a vertical bar (]) as the delimiter, the log11/22/33/44/55 is parsed into the following fields: 11, 22, 33, 44, and 55. You can set the keys toA, B, C, D, and E. If you turn on Incomplete Entry Upload, 55 is uploaded as the value of theD key when Log Service collects the log 11/22/33/55. If you turn off Incomplete Entry Upload, the log11/22/33/55 is dropped because the number of fields parsed from the log does not match the number of the specified keys.
Use System Time	 Specifies whether to use the system time. If you turn on Use System Time, the timestamp of a log indicates the system time of the server when the log is collected. If you turn off Use System Time, you must set the Specify Time Key and Time Format parameters based on the value of the time field that is specified in Extracted Content. For example, if you set the Specify Time Key parameter to time_local and the Time Format parameter to %d/%b/%Y:%H:%M:%S, the timestamp of a log is the value of the time_local field.
Drop Failed to Parse Logs	 Specifies whether to upload logs to Log Service if the logs fail to be parsed. If you turn on Drop Failed to Parse Logs, raw logs that fail to be parsed are not uploaded to Simple Log Service. If you turn off Drop Failed to Parse Logs, raw logs that fail to be parsed are uploaded to Simple Log Service.
Maximum Directory Monitoring Depth	The maximum depth at which the log directory is monitored. Valid values: 0 to 1000. The value 0 indicates that only the specified log file directory is monitored.

7. Optional. Configure parameters in the Advanced Options section. After the settings are complete, click Next.

You can configure advanced settings based on your business requirements. We recommend that you retain the default settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specify whether to enable plug-in processing. If you turn on this switch, you can use plug-ins to process text logs. Note If you turn on Enable Plug-in Processing, the parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.
Upload Raw Log	If you turn on Upload Raw Log , each raw log is uploaded to Simple Log Service as the value of the_ raw _ field together with the log parsed from the raw log.
Topic Generation Mode	 Select the topic generation mode. Null - Do not generate topic: In this mode, the topic field is set to an empty string. When you query logs, you do not need to specify a topic. This is the default value. Machine Group Topic Attributes: In this mode, topics are configured at the machine group level. If you want to distinguish the logs that are generated by different servers, select this mode. File Path RegEx: In this mode, you must specify a regular expression in theCustom RegEx field. The part of a log path that matches the regular expression is used as the topic. If you want to distinguish the logs that are generated by different users or instances, select this mode.
Custom RegEx	Specify a custom regular expression. If you select File Path RegEx for Topic Generation Mode, you must specify a custom regular expression.

Log File Encoding	Select the encoding format of log files. Valid values: utf8 and gbk. • utf8: UTF-8 encoding is used. • gbk: GBK encoding is used.
Timezone	 Select the time zone for the time of logs that are collected. Valid values: System Timezone: If you select this value, the time zone of the server is used. Custom: If you select this value, you must select a time zone based on your business requirements.
Timeout	 If a log file is not updated within the specified period, Logtail considers the log file to be timed out. Valid values: Never: All log files are continuously monitored, and log files never time out. 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select 30 Minute Timeout, you must configure Maximum Timeout Directory Depth. Valid values: 1 to 3.
Filter Configuration	 Specify the conditions to filter logs. Only logs that exactly match the specified filter conditions are collected. Examples: Collect the logs that match the specified filter conditions. If you setKey to level and set Regex to WARNING/ERROR, only logs in which the value of level is WARNING or ERROR are collected. Filter out the logs that do not match the specified filter conditions. If you set Key to level and set Regex to ^?!!*(INFO/DEBUG)).*, logs in which the value of level is INFO or DEBUG are filtered out. If you set Key to url and set Regex to .*^?!.*(healthcheck)).*, logs in which the value of url contains healthcheck are filtered out. For example, if a log contains a field whose key is url and value is /inner/healthcheck/jiankong.html, the log is filtered out.

8. Configure indexes for query and analysis. After you complete the settings, click Next.

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on the collected logs in manual or automatic mode. For more information, see Enable the index feature and configure indexes for a Logstore

⑦ Note

- If you want to guery and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.
- If you configure an index for a field of the Long or Double type, you cannot configure Case Sensitive or Delimiter for the field.

After you complete the settings, you can start to collect logs in delimiter mode.

Appendix: delimiters and sample logs

Logs that are in the delimiter-separated values (DSV) format use line feeds as boundaries. Each line indicates a log. Each log is parsed into multiple fields by using delimiters. Both single-character and multi-character delimiters are supported. If a field contains delimiters, you can enclose the field in a pair of quotes.

Single-character delimiter

The following example shows sample logs with single-character delimiters:

```
05/May/2016:13:30:28,10.10.*.*,"POST /PutData?
 ************ HTTP/1.1",200,18204,aliyun-sdk-java
05/May/2016:13:31:23,10.10.*.*,"POST /PutData?
 ************ HTTP/1.1",401,23472,aliyun-sdk-java
If a log contains single-character delimiters, you must specify the delimiter. You can also specify a quote.
• Delimiter: Available single-character delimiters include the tab character (\t), vertical bar (|), space character, comma (,), and semicolon (;). You
```

can also specify a non-printable character as the delimiter. You cannot specify a double quotation mark (") as the delimiter.

However, a double quotation mark (") can be used as a quote. You can place the double quotation mark at the border of a field, or in the field. If a double quotation mark (") is included in a log field but is not used as a quote, it must be escaped as double quotation marks ("). When Log Service parses log fields, the double quotation marks ("") are automatically converted to a double quotation mark ("). For example, you can specify a comma (,) as the delimiter and a double quotation marks (") are automatically converted to a double quotation mark ("). For example, you can specify a comma (,) as the delimiter and a double quotation mark (") as the quote in a log field. You must enclose the field that contains commas (,) in a pair of quotes. In addition, you must escape the double quotation mark (") in the field to double quotation marks (""). If a processed log is in the **1999,Chevy, "Venture ""Extended Edition, Very Large"", ";,5000.00** format, the log can be parsed into the following five fields: **1999, Chevy, Venture "Extended Edition, Very Large"**, an empty field, and **5000.00**.

• Quote: If a log field contains delimiters, you must specify a pair of quotes to enclose the field. Log Service parses the content that is enclosed in a pair of quotes into a new complete field

Available quotes include the tab character (\t), vertical bar (|), space character, comma (,), semicolon (;), and non-printable characters.

For example, if you specify a comma (,) as the delimiter and a double quotation mark (") as the quote, the log **1997,Ford,E350,"ac, abs,** moon",3000.00 is parsed into the following five fields: **1997, Ford, E350, ac, abs, moon**, and **3000.00**.

Multi-character delimiter

The following example shows sample logs with multi-character delimiters:

05/May/2016:13:30:28&&10.200.**.**&&POST /PutData?

7hAgQ7b1c%3D HTTP/1.1&&200&&18204&&aliyun-sdk-java

05/May/2016:13:31:23&&10.200.**.**&&POST /PutData?

********** HTTP/1.1&&401&&23472&&aliyun-sdk-java

A multi-character delimiter can contain two or three characters, such as ||, &&&, and ^_^. Log Service parses logs based on delimiters. You do not need to use quotes to enclose log fields.

? Note

You must make sure that the delimiters in a field cannot be parsed into a new field. Otherwise, Log Service cannot parse the fields as expected.

For example, if you specify && as the delimiter, the log 1997&&Ford&&E350&&ac&abs&moon&&3000.00 is parsed into the following five fields: 1997, Ford, E350, ac&abs&moon, and 3000.00.

4.3.1.4.5. Collect logs in JSON mode

Log Service allows you to collect JSON logs in JSON mode by using Logtail. After logs are collected, you can transform and ship the logs, and perform multidimensional log analysis. This topic describes how to create a Logtail configuration in JSON mode in the Log Service console to collect logs.

Background information

JSON logs can be written in the object or array structure. A log in the object structure contains key-value pairs, and a log in the array structure contains an ordered list of values.

In JSON mode, Logtail can parse JSON logs in the object structure and extract the keys and values from the first layer of each object. The extracted keys are used as field names, and the extracted values are used as field values. Logtail cannot parse JSON logs in the array structure. If you want to parse JSON logs in the array structure, you can collect data from the JSON logs in full regex or simple mode. For more information, see Collect logs in simple mode or Collect logs in full regex mode.

Sample JSON logs:

{"url": "POST /PutData?

Category=YunOsAccountOpLog&AccessKeyId=U0Ujpek*******&Date=Fri&2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2Bmkd gQ7blc%3D HTTP/1.1", "ip": "10.200.98.220", "user-agent": "aliyun-sdk-java", "request": {"status": "200", "latency": "18204"}, "time": "05/Jan/2020:13:30:28"}

{"url": "POST /PutData?

Category=YunOsAccountOpLog&AccessKeyId=U0Ujpek*******&Date=Fri&2C%2028&20Jun&202013&2006&3A53&3A30&20GMT&Topic=raw&Signature=pD12XYLmGxKQ&2Bmkd gQ7b1c&3D HTTP/1.1", "ip": "10.200.98.210", "user-agent": "aliyun-sdk-java", "request": {"status": "200", "latency": "10204"}, "time": "05/Jan/2020:13:30:29"}

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Import Data section, select JSON Text Log.
- 3. Select the project and Logstore. Then, click Next.
 - You can also click Create Now to create a project and a Logstore.

If you click the plus sign (+) next to **Data Import** below your Logstore to enter the configuration process, the system automatically skips this step. 4. Create a machine group and click **Next**.

Before you create a machine group, make sure that Logtail is installed.

Follow the on-screen instructions to install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows.

After you install Logtail, click **Complete Installation** to create a machine group. If you created a machine group, click **Use Existing Machine Groups**.

5. Select the machine group in the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

() Important

If you apply a machine group immediately after you create the machine group, the heartbeat status of the machine group may be **FAIL**. This issue occurs because the machine group is not connected to Simple Log Service. To resolve the issue, you can click **Automatic Retry**. If the status is still **FAIL**, see What do I do if a Logtail machine group has no heartbeats?

ource Server Groups			Applied Server Groups	
Search by server group name	Q		Search by server group name	Q
2	d			
		>		
		<		
1 Items			0 Items	

6. Create a Logtail configuration and click Next.

The following table describes the Logtail parameters.
Parameter	Description	
Config Name	The name of the Logtail configuration. The name must be 3 to 128 characters in length, and can contain only lowercase letters, digits, hyphens (-), and underscores (_). The name must start and end with a lowercase letter or a digit. ⑦ Note After the Logtail configuration is created, you cannot change the name of the Logtail configuration.	
Log Path	 The directory and name of the log file. You can specify an exact directory and an exact name. You can also use wildcard characters to specify the directory for the log files that match specified conditions. Examples: If you specify /apsara/nuwa/**/*.log, Simple Log Service collects logs from the log files whose names are suffixed by .log in the /apsara/nuwa directory and the recursive subdirectories of the directory. If you specify /var/logs/app_*/*.log, Simple Log Service collects logs from the log files that match the following conditions: The file name contains. Jog. The file is stored in a subdirectory under the/var/logs directory or in a recursive subdirectory of the subdirectory. The name of the subdirectory matches the app_* pattern. Note By default, you can use only one Logtail configuration to collect logs from a log file. If you want to use multiple Logs from the log files is stored in a subdirectory of the directory in which the log file is stored. For example, if you want to use two Logtail configurations to collect logs from a log file, you must create a symbolic link for the directory in which the log file is stored. For example, if you want to use two Logtail configuration and specify the symbolic link in the other Logtail configuration. In -s /home/log/nginx/log /home/log/nginx/link_log When you configure this parameter, you can use only asterisks (*) or question marks (?) as wildcard characters. You can use a question mark (?) to match a single character. 	
Docker File	If you want to collect logs from Docker containers, you can specify the paths and tags of the containers. Logtail monitors the containers to check whether the containers are created or destroyed. Then, Logtail filters the logs of the containers based on tags and collects the filtered logs.	
Mode	The default mode is JSON Mode . You can change the mode.	
Use System Time	 Specifies whether to use the system time. If you turn on Use System Time, the timestamp of a log indicates the system time of the server when the log is collected. If you turn off Use System Time, you must set the Specify Time Key and Time Format parameters based on the time field of JSON logs. For example, if the time information in a JSON log is "time": "05/May/2016:13:30:28", you can set the Specify Time Key parameter to time and the Time Format parameter to %d/%b/%Y:%H:%M:%S. 	
Drop Failed to Parse Logs	 Specifies whether to upload logs to Log Service if the logs fail to be parsed. If you turn on Drop Failed to Parse Logs, raw logs that fail to be parsed are not uploaded to Simple Log Service. If you turn off Drop Failed to Parse Logs, raw logs that fail to be parsed are uploaded to Simple Log Service. 	
Maximum Directory Monitoring Depth	The maximum depth at which the log directory is monitored. Valid values: 0 to 1000. The value 0 indicates that only the specified log file directory is monitored.	

Optional. Configure parameters in the Advanced Options section. After the settings are complete, click Next.
 You can configure advanced settings based on your business requirements. We recommend that you retain the default settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specify whether to enable plug-in processing. If you turn on this switch, you can use plug-ins to process text logs. Note If you turn on Enable Plug-in Processing, the parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.
Upload Raw Log	If you turn on Upload Raw Log , each raw log is uploaded to Simple Log Service as the value of the_ raw_ field together with the log parsed from the raw log.

Topic Generation Mode	 Select the topic generation mode. Null - Do not generate topic: In this mode, the topic field is set to an empty string. When you query logs, you do not need to specify a topic. This is the default value. Machine Group Topic Attributes: In this mode, topics are configured at the machine group level. If you want to distinguish the logs that are generated by different servers, select this mode. File Path RegEx: In this mode, you must specify a regular expression in theCustom RegEx field. The part of a log path that matches the regular expression is used as the topic. If you want to distinguish the logs that are generated by different users or instances, select this mode.
Custom RegEx	Specify a custom regular expression. If you select File Path RegEx for Topic Generation Mode, you must specify a custom regular expression.
Log File Encoding	Select the encoding format of log files. Valid values: utf8 and gbk. • utf8: UTF-8 encoding is used. • gbk: GBK encoding is used.
Timezone	 Select the time zone for the time of logs that are collected. Valid values: System Timezone: If you select this value, the time zone of the server is used. Custom: If you select this value, you must select a time zone based on your business requirements.
Timeout	 If a log file is not updated within the specified period, Logtail considers the log file to be timed out. Valid values: Never: All log files are continuously monitored, and log files never time out. 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select 30 Minute Timeout, you must configure Maximum Timeout Directory Depth. Valid values: 1 to 3.
Filter Configuration	 Specify the conditions to filter logs. Only logs that exactly match the specified filter conditions are collected. Examples: Collect the logs that match the specified filter conditions. If you setKey to level and set Regex to WARNING ERROR, only logs in which the value of level is WARNING or ERROR are collected. Filter out the logs that do not match the specified filter conditions. If you set Key to level and set Regex to ^(?!.*(INFO DEBUG)).*, logs in which the value of level is INFO or DEBUG are filtered out. If you set Key to url and set Regex to .*^(?!.*(healthcheck)).*, logs in which the value of url contains healthcheck are filtered out. For example, if a log contains a field whose key is url and value is /inner/healthcheck/jiankong.html, the log is filtered out.

8. Configure indexes for query and analysis. After you complete the settings, click Next.

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on the collected logs in manual or automatic mode. For more information, see Enable the index feature and configure indexes for a Logstore.

? Note

• If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.

• If you configure an index for a field of the Long or Double type, you cannot configure Case Sensitive or Delimiter for the field.

After you complete the settings, you can start to collect logs in JSON mode.

4.3.1.4.6. Collect logs in NGINX mode

Log Service allows you to collect NGINX logs and perform multidimensional log analysis. This topic describes how to create a Logtail configuration in NGINX mode in the Log Service console to collect logs.

Prerequisites

• A project and a Logstore are created. For more information, see Create a project and Create a Logstore.

• Ports 80 and 443 are enabled for the server from which you want to collect logs.

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Import Data section, select Nginx Text Log.
- 3. Select the project and Logstore. Then, click **Next**. You can also click **Create Now** to create a project and a Logstore.
- If you click the plus sign (+) next to **Data Import** below your Logstore to enter the configuration process, the system automatically skips this step. 4. Create a machine group and click **Next**.

Before you create a machine group, make sure that Logtail is installed. Follow the on-screen instructions to install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows. After you install Logtail, click **Complete Installation** to create a machine group. If you created a machine group, click **Use Existing Machine Groups**.

5. Select the machine group in the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

! Important

If you apply a machine group immediately after you create the machine group, the heartbeat status of the machine group may be **FAIL**. This issue occurs because the machine group is not connected to Simple Log Service. To resolve the issue, you can click **Automatic Retry**. If the status is still **FAIL**, see **What** do I do if a Logtail machine group has no heartbeats?

Source Server Groups			Applied Server Groups	
Search by server group name	Q		Search by server group name	Q
		>		
		<		
1 Items			0 Items	

6. Create a Logtail configuration and click Next.

The following table describes the Logtail parameters.

Parameter	Description
Config Name	The name of the Logtail configuration. The name must be 3 to 63 characters in length, and can contain only lowercase letters, digits, hyphens (-), and underscores (_). The name must start and end with a lowercase letter or a digit. ⑦ Note After the Logtail configuration is created, you cannot change the name of the Logtail configuration.
Log Path	 The directory and name of the log file. You can specify an exact directory and an exact name. You can also use wildcard characters to specify the specified directory for the log files that match specified conditions. Examples: If you specify /apsara/nuwa/**/*.log, Simple Log Service collects logs from the log files whose names are suffixed by .log in the /apsara/nuwa directory and the recursive subdirectories of the directory. If you specify /var/logs/app_*/*.log, Simple Log Service collects logs from the log files that meet the following conditions: The file name contains .log. The file is stored in a subdirectory under the/var/logs directory or in a recursive subdirectory. The name of the subdirectory matches the app_* pattern. Note By default, you can use only one Logtail configuration to collect logs from a log file, you must create a symbolic link for the directory in which the log file is stored. For example, if you want to use two Logtail configurations to collect logs file, you must create a symbolic link for the directory in which the log file is stored. For example, if you want to use two Logtail configurations to collect logs file, you must run the following command to create a symbolic link that points to the directory of the file. Then, you can specify the real path in one Logtail configuration and specify the symbolic link in the other Logtail configuration. In -s /home/log/nginx/log /home/log/nginx/link_log When you configure this parameter, you can use only asterisks (*) or question marks (?) as wildcard characters. You can use a question mark (?) to match a single character.

	If you turn on Blacklist , you must configure a blacklist to exclude specific directories or files from log collection. You can specify exact directories and file names. You can also use wildcard characters to specify directories and file names. Examples:
	 If you select Filter by Directory from a drop-down list in the Filter Type column and enter /home/admin/dir1 for Content, all files in the /home/admin/dir1 directory are skipped.
	 If you select Filter by Directory from a drop-down list in the Filter Type column and enter /home/admin/dir* for Content, the files in all subdirectories whose names are prefixed by dir in the /home/admin/ directory are skipped.
	 If you select Filter by Directory from a drop-down list in the Filter Type column and enter /home/admin/*/dir for Content, all files in dir directories in each subdirectory of the/home/admin/ directory are skipped.
	For example, the files in the /home/admin/a/dir directory are skipped, but the files in the /home/admin/a/b/dir directory are not skipped.
	 If you select Filter by File from a drop-down list in the Filter Type column and enter /home/admin/private*.log for Content, all files whose names are prefixed by private and suffixed by .log in the /home/admin/ directory are skipped.
	 If you select Filter by File from a drop-down list in the Filter Type column and enter /home/admin/private*/*_inner.log for Content, all files whose names are suffixed by _inner.log in the subdirectories whose names are prefixed by private in the /home/admin/ directory are skipped.
Blacklist	For example, the /home/admin/private/app_inner.log file is skipped, but the/home/admin/private/app.log file is not skipped.
	⑦ Note
	 When you configure this parameter, you can use only asterisks (*) or question marks (?) as wildcard characters.
	 You can use an asterisk (*) to match multiple characters.
	You can use a question mark (?) to match a single character.
	 If you use wildcard characters to configure Log Path and you want to skip some directories in the specified directory, you must configure the blacklist and enter a complete directory.
	For example, if you set Log Path to /home/admin/app*/log/*.log and you want to skip all subdirectories in the /home/admin/app1* directory, you must select Filter by Directory and enter /home/admin/app1*/** to configure the blacklist. If you enter/home/admin/app1*, the blacklist does not take effect.
	 When a blacklist is in use, computational overhead is generated. We recommend that you add up to 10 entries to the blacklist.
	 You cannot specify a directory path that ends with a forward slash (/). For example, if you set the path to /home/admin/dir1/, the directory blacklist does not take effect.
Docker File	If you want to collect logs from Docker containers, you can specify the paths and tags of the containers. Logtail monitors the containers to check whether the containers are created or destroyed. Then, Logtail filters the logs of the containers based on tags and collects the filtered logs.
Mode	The default mode is NGINX Configuration Mode. You can change the mode.
	Enter the log configuration section that is specified in the NGINX configuration file. The section starts with log_format . Example:
	<pre>log_format main '\$remote_addr - \$remote_user [\$time_local] "\$request" '</pre>
NGINX Log Configuration	'\$request_time \$request_length ' '\$status \$body_bytes_sent "\$http_referer" '
	<pre>'"\$http_user_agent"';</pre>
	For more information, see Appendix: log formats and sample logs.
NGINX Key	The NGINX keys and values are automatically generated based on the content of NGINX Log Configuration and Log Sample.
	Specifies whether to upload logs to Log Service if the logs fail to be parsed.
Drop Failed to Parse Logs	 If you turn on Drop Failed to Parse Logs, raw logs that fail to be parsed are not uploaded to Simple Log Service.
	 If you turn off Drop Failed to Parse Logs, raw logs that fail to be parsed are uploaded to Simple Log Service.
Maximum Directory Monitoring Depth	The maximum depth at which the log directory is monitored. Valid values: 0 to 1000. The value 0 indicates that only the specified log file directory is monitored.

7. Optional. Configure parameters in the Advanced Options section. After the settings are complete, click Next.

You can configure advanced settings based on your business requirements. We recommend that you retain the default settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specify whether to enable plug-in processing. If you turn on this switch, you can use plug-ins to process text logs. Note If you turn on Enable Plug-in Processing, the parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.

Upload Raw Log	If you turn on Upload Raw Log , each raw log is uploaded to Simple Log Service as the value of the_ raw _ field together with the log parsed from the raw log.
Topic Generation Mode	 Select the topic generation mode. Null - Do not generate topic: In this mode, the topic field is set to an empty string. When you query logs, you do not need to specify a topic. This is the default value. Machine Group Topic Attributes: In this mode, topics are configured at the machine group level. If you want to distinguish the logs that are generated by different servers, select this mode. File Path RegEx: In this mode, you must specify a regular expression in theCustom RegEx field. The part of a log path that matches the regular expression is used as the topic. If you want to distinguish the logs that are generated by different servers.
Custom RegEx	Specify a custom regular expression. If you select File Path RegEx for Topic Generation Mode, you must specify a custom regular expression.
Log File Encoding	 Select the encoding format of log files. Valid values: utf8 and gbk. utf8: UTF-8 encoding is used. gbk: GBK encoding is used.
Timezone	 Select the time zone for the time of logs that are collected. Valid values: System Timezone: If you select this value, the time zone of the server is used. Custom: If you select this value, you must select a time zone based on your business requirements.
Timeout	 If a log file is not updated within the specified period, Logtail considers the log file to be timed out. Valid values: Never: All log files are continuously monitored, and log files never time out. 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select 30 Minute Timeout, you must configure Maximum Timeout Directory Depth. Valid values: 1 to 3.
Filter Configuration	 Specify the conditions to filter logs. Only logs that exactly match the specified filter conditions are collected. Examples: Collect the logs that match the specified filter conditions. If you setKey to level and set Regex to WARNING ERROR, only logs in which the value of level is WARNING or ERROR are collected. Filter out the logs that do not match the specified filter conditions. If you set Key to level and set Regex to ^(?1.*(INFO DEBUG)).*, logs in which the value of level is INFO or DEBUG are filtered out. If you set Key to url and set Regex to .*^(?1.*(healthcheck)).*, logs in which the value of url contains healthcheck are filtered out. For example, if a log contains a field whose key is url and value is /inner/healthcheck/jiankong.html, the log is filtered out.

8. Configure indexes for query and analysis. After you complete the settings, click $\ensuremath{\textbf{Next}}.$

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on the collected logs in manual or automatic mode. For more information, see Enable the index feature and configure indexes for a Logstore.

? Note

- If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.
- If you configure an index for a field of the Long or Double type, you cannot configure Case Sensitive or Delimiter for the field.

After you complete the settings, you can start to collect logs in NGINX mode.

Appendix: log formats and sample logs

Before you collect NGINX access logs, you must specify log_format and access_log in the /etc/nginx/nginx.conf file. The log_format parameter is used to specify the log format. The access_log parameter is used to specify the path in which the NGINX log files are stored.

Log format

The following sample code shows the default values of the log_format and access_log parameters:

log_format main '\$remote_addr - \$remote_user [\$time_local] "\$request" ' '\$request_time \$request_length ' '\$status \$body_bytes_sent "\$http_referer" ' "\$http_user_agent"; access_log /var/logs/nginx/access.log main

The following table describes the log fields.

Log field	Description
remote_addr	The IP address of the client.
remote_user	The username of the client.
time_local	The system time of the server. The value must be enclosed in brackets [].

request	The URI and HTTP protocol of a request.
request_time	The time that is required to process a request. Unit: seconds.
request_length	The length of a request. The length includes the request line, request header, and request body.
status	The status of a request.
body_bytes_sent	The number of bytes in a response that is sent to the client. The size of the response header is excluded.
http_referer	The URL of the source web page.
http_user_agent	The browser information of the client.

Sample log

192.168.1.2 - - [10/Jul/2020:15:51:09 +0800] "GET /ubuntu.iso HTTP/1.0" 0.000 129 404 168 "-" "Wget/1.11.4 Red Hat modified"

4.3.1.4.7. Collect logs in IIS mode

Log Service allows you to collect Internet Information Services (IIS) logs and perform multidimensional log analysis. This topic describes how to create a Logtail configuration in IIS mode in the Log Service console to collect logs.

Prerequisites

- A project and a Logstore are created. For more information, see Create a project and Create a Logstore.
- Ports 80 and 443 are enabled for the server from which you want to collect logs.
- Logs are generated on the server in the IIS, NCSA Common, or W3C Extended format.
- We recommend that you use the W3C Extended log format. If you select the W3C Extended format, you must configure the fields in the W3C Logging Fields dialog box. To do so, you must select **Bytes Sent (sc-bytes)** and **Bytes Received (cs-bytes)** and use the default settings for other fields.



Procedure

- 1. Log on to the Simple Log Service console
- 2. Select a data source.
- In the Import Data section, select IIS Text Log.
- 3. Select the project and Logstore. Then, click Next.

You can also click Create Now to create a project and a Logstore.

If you click the plus sign (+) next to **Data Import** below your Logstore to enter the configuration process, the system automatically skips this step.

4. Create a machine group and click Next.

Before you create a machine group, make sure that Logtail is installed.

Follow the on-screen instructions to install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows.

After you install Logtail, click **Complete Installation** to create a machine group. If you created a machine group, click **Use Existing Machine Groups**.

5. Select the machine group in the Source Server Groups section and move the machine group to the Applied Server Groups section. Then, click Next.

! Important

If you apply a machine group immediately after you create the machine group, the heartbeat status of the machine group may be **FAIL**. This issue occurs because the machine group is not connected to Simple Log Service. To resolve the issue, you can click **Automatic Retry**. If the status is still **FAIL**, see What do I do if a Logtail machine group has no heartbeats?

ource Server Groups		Applied Server Groups	
Search by server group name	Q	Search by server group name	Q
	d		
		>	
		<	
1 Items		0 Items	

6. Create a Logtail configuration.

The following table describes the Logtail parameters.

Parameter	Description	
Config Name	The name of the Logtail configuration. The name must be 3 to 128 characters in length, and can contain only lowercase letters, digits, hyphens (-), and underscores (_). The name must start and end with a lowercase letter or a digit. ⑦ Note After the Logtail configuration is created, you cannot change the name of the Logtail configuration.	
Log Path	 The directory and name of the log file. You can specify an exact directory and an exact name. You can also use wildcard characters to specify the directory and name. For more information, see Wildcard matching. Simple Log Service scans all levels of the specified directory for the log files that match specified conditions. Examples: If you specify /apsara/nuwa/**/*.log. Simple Log Service collects logs from the log files whose names are suffixed by .log in the /apsara/nuwa directory and the recursive subdirectories of the directory. If you specify /var/logs/app_*/*.log. Simple Log Service collects logs from the log files that meet the following conditions: The file name contains. log. The file is stored in a subdirectory under the/var/logs directory or in a recursive subdirectory of the subdirectory. The name of the subdirectory matches the app_* pattern. Note By default, you can use only one Logtail configuration to collect logs from a log file. If you want to use multiple Logtail configurations to collect logs in a log file, you must create a symbolic link for the directory in which the log file is stored. For example, if you want to use two Logtail configuration and specify the symbolic link in the other Logtail configuration and specify the symbolic link in the other Logtail configuration and specify the symbolic link in the other Logtail configuration. In -s /home/log/nginx/log /home/log/nginx/link_log When you configure this parameter, you can use only asterisks (*) or question marks (?) as wildcard characters. You can use a question mark (?) to match a single character. 	
Docker File	If you want to collect logs from Docker containers, you can specify the paths and tags of the containers. Logtail monitors the containers to check whether the containers are created or destroyed. Then, Logtail filters the logs of the containers based on tags and collects the filtered logs. For more information, see Collect container text logs.	

Cloud Defined Storage

Mode	If you have specified IIS - Text Log for the data source, the default mode is IIS Configuration Mode . You can change the mode.		
Log format	 Select the format of logs that are generated on the IIS server. IIS: Microsoft IIS log file format NCSA: NCSA Common log file format W3C: W3C Extended log file format 		
IIS Configuration	 Specify the IIS configuration fields. If you set the Log format parameter to IIS or NCSA, the IIS configuration fields are automatically generated. If you set the Log format parameter to W3C, enter the content that is specified in the dogFile logExtFileFlags field of the IIS configuration file. logExtFileFlags="Date, Time, ClientIP, UserName, SiteName, ComputerName, ServerIP, Method, UriStem, UriQuery, HttpStatus, Win32Status, BytesSent, BytesRecv, TimeTaken, ServerPort, UserAgent, Cookie, Referer, ProtocolVersion, Host, HttpSubStatus" Default path of the IIS6 configuration file: C:\WINDOWS\system32\inetsrv\MetaBase.xml Default path of the IIS7 configuration file: C:\Windows\System32\inetsrv\config\applicationHost.config 		
IIS Key Name	Log Service automatically extracts IIS keys based on the content of IIS Configuration .		
Drop Failed to Parse Logs	 If you turn on Drop Failed to Parse Logs, raw logs that fail to be parsed are not uploaded to Simple Log Service. If you turn off Drop Failed to Parse Logs, raw logs that fail to be parsed are uploaded to Simple Log Service. 		
Maximum Directory Monitoring Depth	The maximum depth at which the log directory is monitored. Valid values: 0 to 1000. The value 0 indicates that only the specified log file directory is monitored.		

7. Optional. Configure parameters in the Advanced Options section. After the settings are complete, click Next.

You can configure advanced settings based on your business requirements. We recommend that you retain the default settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specify whether to enable plug-in processing. If you turn on this switch, you can use plug-ins to process text logs. Note If you turn on Enable Plug-in Processing, the parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.
Upload Raw Log	If you turn on Upload Raw Log , each raw log is uploaded to Simple Log Service as the value of the raw field together with the log parsed from the raw log.
Topic Generation Mode	 Select the topic generation mode. Null - Do not generate topic: In this mode, the topic field is set to an empty string. When you query logs, you do not need to specify a topic. This is the default value. Machine Group Topic Attributes: In this mode, topics are configured at the machine group level. If you want to distinguish the logs that are generated by different servers, select this mode. File Path RegEx: In this mode, you must specify a regular expression in theCustom RegEx field. The part of a log path that matches the regular expression is used as the topic. If you want to distinguish the logs that are generated by different users or instances, select this mode.
Custom RegEx	Specify a custom regular expression. If you select File Path RegEx for Topic Generation Mode, you must specify a custom regular expression.
Log File Encoding	Select the encoding format of log files. Valid values: utf8 and gbk. • utf8: UTF-8 encoding is used. • gbk: GBK encoding is used.
Timezone	 Select the time zone for the time of logs that are collected. Valid values: System Timezone: If you select this value, the time zone of the server is used. Custom: If you select this value, you must select a time zone based on your business requirements.

Timeout	 If a log file is not updated within the specified period, Logtail considers the log file to be timed out. Valid values: Never: All log files are continuously monitored, and log files never time out. 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select 30 Minute Timeout, you must configure Maximum Timeout Directory Depth. Valid values: 1 to 3.
Filter Configuration	 Specify the conditions to filter logs. Only logs that exactly match the specified filter conditions are collected. Examples: Collect the logs that match the specified filter conditions. If you setKey to level and set Regex to WARNING ERROR, only logs in which the value of level is WARNING or ERROR are collected. Filter out the logs that do not match the specified filter conditions. Filter out the logs that do not match the specified filter conditions. Filter out the logs that do not match the specified filter conditions.
	 If you set Key to reven and set Regex to (11. (Int OpEbbo)). , logs in which the value of even is into on DEbbo are filtered out. If you set Key to url and set Regex to .*^(?!.*(healthcheck)).*, logs in which the value of url contains healthcheck are filtered out. For example, if a log contains a field whose key is url and value is /inner/healthcheck/jiankong.html, the log is filtered out.

8. Configure indexes for query and analysis. After you complete the settings, click Next.

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on the collected logs in manual or automatic mode. For more information, see Enable the index feature and configure indexes for a Logstore.

? Note

- If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.
- If you configure an index for a field of the Long or Double type, you cannot configure Case Sensitive or Delimiter for the field.

After you complete the settings, you can start to collect logs in IIS mode.

Appendix: sample logs and field descriptions

The following example shows a sample IIS log:

#Software: Microsoft Internet Information Services 7.5

- #Version: 1.0 #Date: 2020-09-08 09:30:26

#Fields: date time s-sitename s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) sc-status sc-substatus sc-win32-s tatus sc-bytes cs-bytes time-taken

2009-11-26 06:14:21 W3SVC692644773 125.67.67.* GET /index.html - 80 - 10.10.10.10 Baiduspider+(+http://www.baidu.com)200 0 64 185173 296 0

• Field prefixes

Prefix	Description
S-	The server action.
C-	The client action.
CS-	The client-to-server action.
SC-	The server-to-client action.

• Fields

Log field	Description	
date	The date on which the client sends the request.	
time	The point in time at which the client sends the request.	
s-sitename	The Internet service name and instance ID of the site that is visited by the client.	
s-computername	The name of the server on which the log is generated.	
s-ip	The IP address of the server on which the log is generated.	
cs-method	The HTTP request method that is used by the client, for example, GET or POST.	
cs-uri-stem	The URI resource that is requested by the client.	

cs-uri-query	The query string that follows the question mark (?) in the HTTP request.	
s-port	The port number of the server.	
cs-username	 The authenticated domain name or username that is used by the client to access the server. Authenticated users are referenced as domain\username. Anonymous users are indicated by a hyphen (-). 	
c-ip	The real IP address of the client that sends the request.	
cs-version	The protocol version that is used by the client, for example, HTTP 1.0 or HTTP 1.1.	
cs(User-Agent)	The browser that is used by the client.	
Cookie	The content of the sent or received cookie. If no cookie is sent or received, a hyphen (-) is displayed.	
referer	The previous site that is visited by the user.	
cs-host	The host information.	
sc-status	The HTTP status code that is returned by the server.	
sc-substatus	The HTTP substatus code that is returned by the server.	
sc-win32-status	The Windows status code that is returned by the server.	
sc-bytes	The number of bytes that are sent by the server.	
cs-bytes	The number of bytes that are received by the server.	
time-taken	The time that is required to process the request. Unit: milliseconds.	

4.3.1.4.8. Collect logs in Apache mode

Log Service allows you to collect Apache logs and perform multidimensional log analysis. This topic describes how to create a Logtail configuration in Apache mode in the Log Service console to collect logs.

Prerequisites

- A project and a Logstore are created. For more information, see Create a project and Create a Logstore.
- Ports 80 and 443 are enabled for the server from which you want to collect logs.
- The print format, log path, and log file name are specified in the Apache configuration file. For more information, see Appendix: log formats and

Procedure

- 1. Log on to the Simple Log Service console
- 2. Select a data source.
- In the Import Data section, select Apache Text Log.
- 3. Select the project and Logstore. Then, click Next.
 - You can also click **Create Now** to create a project and a Logstore.
- If you click the plus sign (+) next to **Data Import** below your Logstore to enter the configuration process, the system automatically skips this step. 4. Create a machine group and click **Next**.
 - Before you create a machine group, make sure that Logtail is installed.
 - Follow the on-screen instructions to install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows.

After you install Logtail, click **Complete Installation** to create a machine group. If you created a machine group, click **Use Existing Machine Groups**.

5. Select the machine group in the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

! Important

If you apply a machine group immediately after you create the machine group, the heartbeat status of the machine group may be **FAIL**. This issue occurs because the machine group is not connected to Simple Log Service. To resolve the issue, you can click **Automatic Retry**. If the status is still **FAIL**, see What do I do if a Logtail machine group has no heartbeats?

Source Server Groups			Applied Server Groups	
Search by server group name	Q		Search by server group name	Q
<mark>/</mark>	d			
		>		
		<		
1 Items			0 Items	

6. Create a Logtail configuration.

The following table describes the Logtail parameters.

Parameter	Description		
Config Name	The name of the Logtail configuration. The name must be 3 to 128 characters in length, and can contain only lowercase letters, digits, hyphens (-), and underscores (_). The name must start and end with a lowercase letter or a digit. Items		
Log Path	 The directory and name of the log file. The specified log file name can be a complete file name or a file name that contains wildcards. For more information, see Wildcard matching. Log Services scans all levels of the specified directory to match log files. Examples: If you specify /apsara/nuwa/***.log, Log Service matches the files whose name is suffixed by.log in the /apsara/nuwa directory and its recursive subdirectories. If you specify /var/log/sapp_***.log, Log Service matches the files that meet the following conditions: The file name contains .log. The file is stored in a subdirectory of the/var/logs directory or in a recursive subdirectory of the subdirectory and the subdirectory matches the app_* pattern. Note By default, each log file can be collected by using only one Logtail configuration. To use multiple Logtail configurations to collect one log file, we recommend that you create a symbolic link that points to the directory where the file is located. For example, you want to collect two copies of the log.log file from the /home/log/inginx/log/log.log directory. When you configure the Logtail configurations, use the original path in one Logtail configuration and use the symbolic link in the other Logtail configuration. 		
	 In -s /home/log/nginx/log /home/log/nginx/link_log You can use only asterisks (*) and question marks (?) as wildcards in the log path. 		
Docker File	Logtail monitors the containers to check whether the containers are created or destroyed. Then, Logtail filters the logs of the containers based on tags and collects the filtered logs. For more information about container text logs, see <u>Collect container text logs</u> .		
Mode	If you have specified Apache - Text Log for the data source, the default mode is Apache Configuration Mode . You can change the mode.		
Log format	Select a log format based on the format specified in the Apache configuration file. Valid values: common, combined, and Custom.		
APACHE Logformat Configuration	Enter the log configuration section that is specified in the Apache configuration file. The section starts with LogFormat. For more information, see Appendix: log formats and sample logs. • If you set Log format to common or combined, the system automatically inserts a value into this field. Check whether the value is the same as the value specified in the Apache configuration file. • If you set Log format to Custom, specify a value based on your business requirements. For example, you can enter LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %D %f %k %p %q %R %T %l %O" customized.		

APACHE Key Name	Log Service automatically reads Apache keys from the value of the APACHE Logformat Configuration field.	
Drop Failed to Parse Logs	 Specifies whether to drop logs that fail to be parsed. If you turn on Drop Failed to Parse Logs, logs that fail to be parsed are not uploaded to Log Service. If you turn off Drop Failed to Parse Logs, raw logs are uploaded to Log Service if the logs fail to be parsed. 	
Maximum Directory Monitoring Depth	The maximum depth at which the log directory is monitored. Valid values: 0 to 1000. The value 0 indicates that only the specified log file directory is monitored.	

7. Optional. Configure parameters in the Advanced Options section. After the settings are complete, click Next.

You can configure advanced settings based on your business requirements. We recommend that you retain the default settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description			
	Specify whether to enable plug-in processing. If you turn on this switch, you can use plug-ins to process text logs.			
Enable Plug-in Processing	Note If you turn on Enable Plug-in Processing, the parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.			
Upload Raw Log	If you turn on Upload Raw Log , each raw log is uploaded to Simple Log Service as the value of the _raw_ field together with the log parsed from the raw log.			
Topic Generation Mode	 Select the topic generation mode. Null - Do not generate topic: In this mode, the topic field is set to an empty string. When you query logs, you do not need to specify a topic. This is the default value. Machine Group Topic Attributes: In this mode, topics are configured at the machine group level. If you want to distinguish the logs that are generated by different servers, select this mode. File Path RegEx: In this mode, you must specify a regular expression in theCustom RegEx field. The part of a log path that matches the regular expression is used as the topic. If you want to distinguish the logs that are generated by different users or instances, select this mode. 			
Custom RegEx	Specify a custom regular expression. If you select File Path RegEx for Topic Generation Mode, you must specify a custom regular expression.			
Log File Encoding	 Select the encoding format of log files. Valid values: utf8 and gbk. utf8: UTF-8 encoding is used. gbk: GBK encoding is used. 			
Timezone	 Select the time zone for the time of logs that are collected. Valid values: System Timezone: If you select this value, the time zone of the server is used. Custom: If you select this value, you must select a time zone based on your business requirements. 			
Timeout	 If a log file is not updated within the specified period, Logtail considers the log file to be timed out. Valid values: Never: All log files are continuously monitored, and log files never time out. 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select 30 Minute Timeout, you must configure Maximum Timeout Directory Depth. Valid values: 1 to 3. 			
Filter Configuration	 Specify the conditions to filter logs. Only logs that exactly match the specified filter conditions are collected. Examples: Collect the logs that match the specified filter conditions. If you setKey to level and set Regex to WARNING ERROR, only logs in which the value of level is WARNING or ERROR are collected. Filter out the logs that do not match the specified filter conditions. If you set Key to level and set Regex to ^(?!.*(INFO DEBUG)).*, logs in which the value of level is INFO or DEBUG are filtered out. If you set Key to url and set Regex to .*^(?!.*(InFo DEBUG)).*, logs in which the value of url contains healthcheck are filtered out. For example, if a log contains a field whose key is url and value is /inner/healthcheck/jiankong.html, the log is filtered out. 			

8. Configure indexes for query and analysis. After you complete the settings, click Next.

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on the collected logs in manual or automatic mode. For more information, see Enable the index feature and configure indexes for a Logstore.

? Note

- If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.
- If you configure an index for a field of the Long or Double type, you cannot configure Case Sensitive or Delimiter for the field.

After you complete the settings, you can start to collect logs in Apache mode.

Appendix: log formats and sample logs

Before you collect Apache logs, you must specify the print format, log path, and log file name. For example, **CustomLog** "/var/log/apache2/access_log" combined indicates that the combined format is used when logs are printed. The log file path is /var/log/apache2/access_log. Log Service supports the following log formats. A sample log is also provided.

- Log formats
 - The combined log format:

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined

• The common log format:

LogFormat "%h %l %u %t \"%r\" %>s %b"

• A custom log format:

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %D %f %k %p %q %R %T %I %O" customized

The following table describes the related log fields. For more information, see mod_log_config.

Format string	Log field	Description
%a	client_addr	The IP address of the client.
%A	local_addr	The local IP address.
%b	response_size_bytes	The number of bytes in a response. If no bytes are sent, a hyphen (-) is displayed for this field.
%В	response_bytes	The number of bytes in a response. If no bytes are sent, the digit 0 is displayed for this field.
%D	request_time_msec	The time required to process a request. Unit: microseconds.
%f	filename	The file name.
%h	remote_addr	The name of the remote host.
%Н	request_protocol_supple	The request protocol.
%1	bytes_received	The number of bytes that are received by the server. This field is recorded in logs only after you enable the mod_logio module.
%k	keep_alive	The number of keep-alive requests handled on the connection.
%I	remote_ident	The information that is provided by the remote host for identification.
%m	request_method_supple	The HTTP request method.
%0	bytes_sent	The number of bytes that are sent by the server. This field is recorded in logs only after you enable the mod_logio module.
%р	remote_port	The port number of the server.
%P	child_process	The ID of the child process.
%q	request_query	The query string. If no query strings exist, an empty string is displayed.
%r	request	The content of the request. The content includes the method name, address, and HTTP protocol.
%R	response_handler	The type of the handler that generates a response on the server.

%s	status	The initial HTTP status of a response.
%>s	status	The final HTTP status of a response.
%t	time_local	The point in time at which the server receives a request.
%T	request_time_sec	The time required to process a request. Unit: seconds.
%u	remote_user	The username of the client.
%U	request_uri_supple	The URI in a request. The URI does not include the query string.
%v	server_name	The name of the server.
%V	server_name_canonical	The name of the server. The name is specified by using the UseCanonicalName directive.
"%{User-Agent}i"	http_user_agent	The information of the client.
"%{Rererer}i"	http_referer	The URL of the web page. The URL is linked to the resource that is being requested.

Sample log

192.168.1.2 - - [02/Feb/2020:17:44:13 +0800] "GET /favicon.ico HTTP/1.1" 404 209 "http://localhost/x1.html" "Mozilla/5.0 (Macintosh; Intel M ac OS X 10_11_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.97 Safari/537.36"

4.3.1.4.9. Configure parsing scripts

When Log Service collects logs, Log Service extracts some fields in raw logs as log content based on specific parsing methods. This way, you can collect logs based on your business requirements. This topic describes the parsing methods that are supported by Log Service.

Specify a method to separate log lines

A complete access log such as an NGINX access log occupies a line. Separate multiple logs with line feeds. The following example shows two access logs:

203.0.113.10 - - [13/Mar/2016:10:00:10 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)" 203.0.113.10 - - [13/Mar/2016:10:00:11 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)"

In most cases, logs for Java applications contain multiple lines. Therefore, logs are separated based on the start part in the first line of a log. The following example shows a Java application log:

[2016-03-18T14:16:16,000] [INFO] [SessionTracker] [SessionTrackerImpl.java:148] Expiring sessions 0x152436b9a12aed2, 50000 0x152436b9a12aed2, 50000 0x152436b9a12aed1, 50000 0x152436b9a12aed0, 50000

The preceding Java application log starts with a datetime value. To ensure the accuracy of log collection, you can specify a regular expression to match the start part in the first line of a log. In this example, the regular expression that is used to match the start part in the first line of a log is $\frac{1}{1+1-1}$. The following figure shows how to enter a sample log and specify a regular expression in the Log Service console.

Figure 1. Full regex mode

Mode:	Full Regex Mode V
* Singleline :	Single line mode means every row contains only one log. For cross-row logs (such as Java stack logs), disable the single line mode and set a regular expression.
* Log Sample:	[2016-03-18T14:16:16,000] [INFO] [SessionTracker] [SessionTrackerImpl Java:148] Expiring sessions 0x152436b9a12aect, 50000 0x152436b9a12aed2, 50000 0x152436b9a12aed0, 50000 0x152436b9a12aed0, 50000
* Regex to Match First	*.el/ld+-ld+-ld+-ld+-ld+-ld+-ld+-ld+-ld+-ld+-
Line:	Matched Items:1 The automatically generated results are only for reference. You can also Manual

Extract log fields

A log contains one or more key-value pairs based on the data model of Log Service. If you want to extract specific fields for analysis, you must specify a regular expression to match the content that you want to extract. If you do not need to process the content of a log, you can process the log as a key-value pair.

The following example shows two access logs. You can use one of the following two methods to parse the logs.

203.0.113.10 - - [13/Mar/2016:10:00:10 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)"

203.0.113.10 - - [13/Mar/2016:10:00:11 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)"

• Extract specific fields.

• Extract all.

In this example, the regular expression is (.*) . The extracted content is 203.0.113.10 - - [13/Mar/2016:10:00:10 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)" .

Specify the log time

A log must contain a time field whose value is a UNIX timestamp based on the data model of Log Service. You can use the system time when Logtail collects a log or the time in the log content as the log time.

The following example shows two access logs. You can use one of the following two methods to parse the logs.

203.0.113.10 - - [13/Mar/2016:10:00:10 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se

203.0.113.10 - - [13/Mar/2016:10:00:11 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)"

• Extract the time in the log content as the log time.

In this example, the time in the log content is 13/Mar/2016:10:00:10 . To extract the time, use the following time expression: %d/%b/%Y:%H:%M:%s . • Use the system time when the log was collected by Logtail as the log time.

If you use the system time when the log was collected by Logtail as the log time, the time is converted to a timestamp.

4.3.1.4.10. Time formats

When you use Logtail to collect logs, you must specify time formats based on the time strings of raw logs. Logtail extracts a time string from a raw log and parses the string into a UNIX timestamp. This topic describes the commonly used time formats and provides related examples.

Commonly used time formats of logs

The following table describes the time formats that are supported by Logtail.

? Note

- The timestamp of a log in Log Service is accurate to seconds. Therefore, you can specify the time format only to seconds.
- You need to specify the time format only for the time in a time string. Other parameters such as the time zone are not required.
- In Linux, Logtail supports all time formats provided by the strftime function. Logtail can parse and use all log time strings that can be formatted by using the strftime function.

Format	Description	Example

%a	The abbreviated day name.	Fri
%A	The full day name.	Friday
%b	The abbreviated month name.	Jan
%B	The full month name.	January
%d	The day of a month. Valid values: 01 to 31.	07, 31
%h	The abbreviated month name. The format is equivalent to $\% b$.	Jan
%Н	The hour in the 24-hour format.	22
%I	The hour in the 12-hour format.	11
%m	The month. Valid values: 01 to 12.	08
%M	The minute. Valid values: 00 to 59.	59
%n	The line feed.	A line feed
%р	The abbreviation that indicates the morning or afternoon. Valid values: AM and PM.	AM or PM
%r	The time in the 12-hour format. The format is equivalent to %1:%M:%S %p.	11:59:59 AM
%R	The time expressed in hours and minutes. The format is equivalent to %H:%M .	23:59
%S	The second. Valid values: 00 to 59.	59
%t	The tab character.	None
%у	The two-digit year number. Valid values: 00 to 99.	04 or 98
%Y	The four-digit year number.	2004 or 1998
%C	The two-digit century number. Valid values: 00 to 99.	16
%e	The day of a month. Valid values: 1 to 31. Prefix a single-digit number with a space character.	7 or 31
%j	The day of a year. Valid values: 001 to 366.	365
%u	The day of a week as a number. Valid values: 1 to 7. The value 1 indicates Monday.	2
%U	The week of a year. Sunday is the first day of each week. Valid values: 00 to 53.	23
%V	The week of a year. Monday is the first day of each week. Valid values: 01 to 53. If a week that contains January 1 has four or more January days, the week is the first week of a year. Otherwise, the next week is considered the first week of a year.	24
%w	The day of a week as a number. Valid values: 0 to 6. The value 0 indicates Sunday.	5
%W	The week of a year. Monday is the first day of each week. Valid values: 00 to 53.	23

%c	The date and time in the ISO 8601 format.	Tue Nov 20 14:12:58 2020
%x	The date in the ISO 8601 format.	Tue Nov 20 2020
%X	The time in the ISO 8601 format.	11:59:59
%s	The UNIX timestamp.	1476187251

Examples

The following table lists commonly used time formats. It also provides related examples and time expressions.

Example	Time expression	Time format
2017-12-11 15:05:07	%Y-%m-%d %H:%M:%S	Custom
[2017-12-11 15:05:07.012]	[%Y-%m-%d %H:%M:%S]	Custom
02 Jan 06 15:04 MST	%d %b %y %H:%M	RFC822
02 Jan 06 15:04 -0700	%d %b %y %H:%M	RFC822Z
Monday, 02-Jan-06 15:04:05 MST	%A, %d-%b-%y %H:%M:%S	RFC850
Mon, 02 Jan 2006 15:04:05 MST	%A, %d %b %Y %H:%M:%S	RFC1123
2006-01-02T15:04:05Z07:00	%Y-%m-%dT%H:%M:%S	RFC3339
2006-01-02T15:04:05.999999999207:00	%Y-%m-%dT%H:%M:%S	RFC3339Nano

4.3.1.4.11. Import historical log files

This topic describes how to import historical log files from a server to Log Service. By default, Logtail collects only incremental logs from servers. You can configure Logtail to collect historical logs.

Prerequisites

- Logtail V0.16.15 (Linux), Logtail V1.0.0.1 (Windows), or later is installed on the server. For more information, see Install Logtail on a Linux server and Install Logtail on a Windows server.
- A Logtail configuration is created and applied to a machine group. For more information, see Configure text log collection. If you use the Logtail configuration to import only historical files, you can specify a log collection path that does not exist.

Background information

Logtail collects logs based on the modifications in the log files that are monitored. Logtail can also collect logs by loading events from local files. Logtail collects historical logs by loading local events.

You must import historical log files to the installation directory of Logtail. The directory varies based on the operating system.

- Linux: /usr/local/ilogtail
- Windows:
- 32-bit: C:\Program Files\Alibaba\Logtail
- 64-bit: C:\Program Files (x86)\Alibaba\Logtail

? Note

- The maximum interval between the time when a local event is generated and the time when the local event is imported is 1 minute.
- If a local event is loaded, Logtail sends the LOAD_LOCAL_EVENT_ALARM message to the server.
- If you want to import a large number of log files, we recommend that you modify the startup parameters of Logtail to increase the value of the cpu_usage_limit parameter to 2 or more and increase the value of the mem_usage_limit parameter to 512 MB or more. For more information, see Configure the startup parameters of Logtail.

Procedure

1. Obtain the unique identifier of the Logtail configuration.

Open the user_log_config.json file in the directory where Logtail is installed. You can obtain the unique identifier of the Logtail configuration from this file.

For example, to obtain the unique identifier of the Logtail configuration in a Linux server, run the following command:

grep "##" /usr/local/ilogtail/user_log_config.json | awk '{print \$1}'

"##1.0##log-config-test\$multi"

- "##1.0##log-config-test\$ecs-test"
- "##1.0##log-config-test\$metric_system_test"
- "##1.0##log-config-test\$redis-status"

Cloud Defined Storage

2. Add a local event.

- i. Create the local_event.json file in the Logtail installation directory.
- ii. Add the local event in the JSON format to the local_event.json file of the Logtail installation directory. The following example shows the format of the local event:

I f
"config": "\${your config unique id}",
"dir" : "\${your_log_dir}",
"name" : "\$(your_log_file_name)"
),
(
}
1
⑦ Note

To prevent Logtail from loading invalid JSON files, we recommend that you save the configurations of the local event to a temporary file. Then, edit and copy the configurations to the local_event.json file.

Parameter	Description
config	Enter the unique identifier that is obtained in Step 1. Example: ##1.0##log- config-test\$ecs-test.
dir	The directory in which historical log files are saved. Example:/data/logs. O Note The directory cannot end with a forward slash (/).
name	The name of the historical log file. The name can contain wildcards. Example: access.log.2018-08-08 and access.log*.

The following example shows how to configure a local event in Linux by using the cat /usr/local/ilogtail/local_event.json command:

FAQ

How do I check whether Logtail loads a Logtail configuration?

After you save the local_event.json file, Logtail loads the configurations of the local event to the memory within 1 minute. Then, the content of the local_event.json file is deleted.

You can use the following methods to check whether the Logtail configuration is loaded.

- i. If no content exists in the local_event.json file, Logtail reads the local event from the file.
- ii. Check whether the ilogtail.LOG file in the Logtail installation directory contains the **process local event** parameter. If the content in the local_event.json file is cleared but the **process local event** parameter does not exist, the content of the local_event.json file may be invalid and filtered out.
- Why am I unable to collect a log file after a Logtail configuration is loaded?
- The Logtail configuration is invalid.
- $\circ~$ The configurations of the local event in the local_event.json file are invalid.
- $\circ\;$ The log file does not exist in the path that is specified in the Logtail configuration.
- The log file has been collected by Logtail.

4.3.1.4.12. Log topics

Logs can be identified by log topics. When you collect logs, you can specify a topic for the logs.

You can specify a topic in the following scenarios: when you use Logtail to collect logs, when you call API operations, or when you use an SDK to upload log data. In the Log Service console, you can set the topic generation mode to Null - Do not generate topic, Machine Group Topic Attributes, or File Path RegEx.

- Null Do not generate topic
- In this mode, the topic is an empty string. You can query logs without the need to specify a topic.
- Machine Group Topic Attributes

You can use this mode to identify logs that are generated on different servers. If the logs are saved with the same file name or the logs are saved in the same directory, you can specify different topics to identify the logs.

You can add servers to different machine groups, and configure different topic attributes for the machine groups. When you create a Logtail configuration, set **Topic Generation Mode** to **Machine Group Topic Attributes**. If Logtail sends the logs of a server in a machine group to Log Service, Logtail uploads the topic attributes of the machine group as topic names. You can use the topic attributes as filters to query logs.

• File Path RegEx

You can use this mode to identify logs that are generated by different users or instances. Log Service stores logs in different directories for different users or instances. However, if duplicate sub-directory names or log file names exist in these directories, Log service cannot identify which user or instance generates the logs.

To resolve this issue, you can set **Topic generation modes** to **File Path Regex** and enter the regular expression of the log file path when you create a Logtail configuration. The regular expression must match the log file path. When Logtail sends logs to Log Service, Logtail uploads the username or the instance name as the topic name. You can use the topic name as a filter to query logs.

Logs that are generated by different users or instances may be stored in different files with the same name. Each file is stored in a different directory. For example, three log files are all named service.log and you only specify the service.log file in the /logs directory as the log source when you collect logs from these files. After the logs are sent to Log Service, Log Service cannot identify which users or instances generate the logs. To resolve this issue, you can set **Topic Generation Mode** to **File Path RegEx**, and enter the $(\cdot, *)/(serviceA/...*)$ regular expression. Then, Log Service generates the following topics for logs that are in different directories: userA, userB, and userC.

100	JS	
1	-	/userA/serviceA
	1	- service.log
1	-	/userB/serviceA
	1	- service.log
1	-	/userC/serviceA
	1	- service.log

? Note

You must escape the forward slash (/) in the regular expression that is used to match file paths.

To extract multiple fields from a file path, you can use the 2P<key> sub-expression to extract fields from the layers of the file path. The value of the key parameter can only contain lowercase letters and digits. Example:

 $\label{eq:loss_log} $$ \hfill $$ \$

The following custom tags are created for logs:

"__tag__ : service : serviceA" "__tag__ : user : userB"

· Static topic generation

? Note

You can set Topic Generation Mode to File Path RegEx. In the Custom RegEx field, enter customized:// + custom topic name .

Static topic generation is supported by Logtail V0.16.21 (Linux) and later. 4.3.1.5. Collect container logs

4.3.1.5.1. Overview

Log Service allows you to collect Kubernetes container logs in DaemonSet mode or Sidecar mode. This topic describes the procedures and differences of log collection in the two modes.

Log collection modes

Log collection in DaemonSet mode features simple O&M, low resource usage, and flexible configurations. You can collect container stdout and stderr. You can also collect container text logs. In DaemonSet mode, Logtail collects logs from all containers on the DaemonSet-specific node. However, in this mode, performance bottleneck issues may occur on Logtail, and containers are loosely isolated. In Sidecar mode, a Sidecar container is created for each container from which you want to collect logs. In this mode, Logtail provides good performance, and tenants are completely isolated.

Log collection configurations

You can create log collection configurations by using the Log Service console or custom resource definitions (CRDs). The following table describes the differences between the two modes.

Item	CRD	Log Service console
Operation complexity	Low	Moderate
Feature diversity	All configurations that the console supports and advanced configurations that the console does not support	Moderate
Ease of use	Moderate	Low
Network connection	Connected to a Kubernetes cluster	Connected to the Internet
Integration with container applications	Supported	Not supported

Authentication method	Kubernetes authentication	Authentication based on Alibaba Cloud accounts

Log collection procedures

The following procedure describes how to collect logs in DaemonSet mode:

- 1. Install the Logtail component.
- 2. Create a log collection configuration.

Log Service allows you to create log collection configurations by using CRDs or the Log Service console to collect container logs from Kubernetes clusters.

- Use CRDs to collect container logs in DaemonSet mode.
- Use the Log Service console to collect container text logs in DaemonSet mode.
- Use the Log Service console to collect container stdout and stderr in DaemonSet mode.

? Note

If you use CRDs, resources such as projects, Logstores, indexes, machine groups, and Logtail configurations are automatically created. In addition, this method leads to better integration with Kubernetes. We recommend that you use this method. If you use the Log Service console, you need to only perform simple operations. The first time you use Log Service to collect container logs, we recommend that you use this method.

The following procedure describes how to collect logs in Sidecar mode:

- 1. Install the Logtail component.
- 2. Install Sidecar and create a log collection configuration.

Log Service allows you to create log collection configurations by using CRDs or the Log Service console to collect container logs from Kubernetes clusters.

- Use CRDs to collect container text logs in Sidecar mode.
- Use the Log Service console to collect container text logs in Sidecar mode.

4.3.1.5.2. Install the Logtail component

This topic describes how to install the Logtail component in a Kubernetes cluster.

Background information

Before you can collect container logs from a Kubernetes cluster, you must install the Logtail component.

When you install the Logtail component, the following operations are automatically completed:

- 1. The alibaba-log-configuration ConfigMap is created. This ConfigMap stores the configuration information about Log Service, such as project information.
- 2. Optional. The AliyunLogConfig custom resource definition (CRD) is created.
- 3. Optional. The alibaba-log-controller Deployment is created. This Deployment is used to monitor the changes in the AliyunLogConfig CRD and create Logtail configurations.
- 4. The logtail-ds DaemonSet is created. This DaemonSet is used to collect logs from nodes.

Alibaba Cloud Container Service for Kubernetes (ACK) clusters

You can install the Logtail component in an existing ACK cluster. You can also install the Logtail component when you create an ACK cluster. To install the Logtail component when you create an ACK cluster, you must select **Enable Log Service**.

Install the Logtail component in an existing ACK cluster

- 1. Log on to the Log Service console
- 2. In the left-side navigation pane, click Clusters.
- 3. On the **Clusters** page, find and click the cluster in which you want to install the Logtail component.
- 4. In the left-side navigation pane of the page that appears, choose **Operations > Add-ons**
- 5. On the Logs and Monitoring tab, find the logtail-ds component and click Install.

After the component is installed, a machine group named k@s-group-\${your_k@s_cluster_id} and a Logstore named config-operation-log are automatically created in the project that you use.

() Important Do not delete the config-operation-log Logstore.

Install the Logtail component when you create an ACK cluster

- 1. Log on to the Log Service console
- 2. In the left-side navigation pane, click **Clusters**.
- 3. On the Clusters page, click Create Kubernetes Cluster.
- 4. In the Component Configurations step, select Enable Log Service.

```
? Note
```

In this example, only the steps that are required to enable Log Service are provided. For more information about how to create an ACK cluster, see Create a Kubernetes cluster in Container Service for Kubernetes User Guide.

If you select **Enable Log Service**, the system prompts you to create a Log Service project. You can use one of the following methods to create a project:

• Select Project

You can select an existing project to manage the container logs that are collected.

1. 2.

3. 4.

		-					
	Log Service	Enable Log Service SP	ricing Details			a	
		Select Project	Create Project	k8s-log-c1t	7da3daed3	▼ 0	
0	Create Project						
	Log Service automatical the unique ID of the ACk	lly creates a project nam K cluster that is created.	ed k8s-log-{ClusterII	to manage the co	ontainer logs that are col	lected. ClusterID	indicates
	Log Service	✓ Enable Log Service	Pricing Details				
		Select Project	Create Project				
		A project named k8s-log-{Cl	lusterID} will be automatically	created.			
Af at	ter the component is ins utomatically created in th	stalled, a machine group he project that you use.	named k8s-group-\${y	our_k8s_cluster_id}	and a Logstore named	config-operation	-log are
	! Important						
	Do not delete the conf:	ig-operation-log Logst	ore.				
Se	lf-managed Kube	ernetes clusters					
1. La 2. Ci	g on to the Log Service con reate a project whose na	sole	-custom-				
E. C.	ample: k8s-log-custom-	sd89ehdg. For more info	rmation, see Create a	project.			
3. Lo	og on to your Kubernetes	s cluster.					
4. Ri	un the following commar	nd to install the alibaba-l	og-controller componer	nt.			
	! Important						
	Make sure that the kubectl command-line tool is installed on the machine on which you want to run the command.						
ı. ii.	Upload the script to the	machine.	JL.				
iii.	Go to the directory of th	e script and modify pern	nissions.				
	chmod 744 ./alicloud-1	log-k8s-custom-install.s	h;				
iv.	Run the following install	lation command:					
	./alicloud-k8s-log-ins ccess-key-id} {access-	staller.shcluster-id -key-secret}	\${your_k8s_cluster_id}	ali-uid \${your_al	i_uid}region-id \${yo	ur_k8s_cluster_rec	gion_id} {a
	You can configure the pa	arameters in the comma	nd based on your busir	ness requirements. Th	ne following table describ	pes the parameters	s.
	Parameter		Description				
	your_k8s_cluster_id		The ID of your Kuberne	tes cluster.			
			The ID of your Alibaba	Cloud account.			
			② Note				
	your_ali_uid		The ID of an Alibaba Alibaba Cloud accourt	Cloud account is a string nt, see <mark>Configure an acc</mark>	g. For more information abo ount ID on a server.	ut how to view the ID	of an
	your_k8s_cluster_regio	on_id	The ID of the region wh	ere your Kubernetes clu	ster resides.		
	access-key-id		The AccessKey ID of yo	ur Alibaba Cloud accour	t. For more information, see	Obtain an AccessKey	ı pair.
	access-key-secret		The AccessKey secret o	of your Alibaba Cloud acc	count. For more information	, seeObtain an Access	sKey pair.
	After the component is i automatically created in	installed, a machine grount the project that you use	up named k8s-group-\$	{your_k8s_cluster_id	and a Logstore name	config-operati	on-log are
	() Important						
	. important						

- Do not delete the config-operation-log Logstore.
- If you install the component in a self-managed Kubernetes cluster, Logtail is granted the privileged permissions. This helps prevent the container text file busy error that occurs when other pods are deleted. For more information, see Bug 1468249, Bug 1441737, and Issue 34538.

FAQ

• How do I collect and send container logs from multiple Kubernetes clusters to the same Log Service project?

• Alibaba Cloud ACK clusters

If you want to collect and send container logs from multiple ACK clusters to the same Log Service project, you must select the same project when you create the ACK clusters.

Self-managed Kubernetes clusters

If you want to collect and send container logs from multiple self-managed Kubernetes clusters to the same Log Service project, you must set the **{your-project-suffix}** parameter to the same value when you install the Logtail component in each of the Kubernetes clusters.

```
? Note
```

You can collect and send container logs from multiple self-managed Kubernetes clusters to the same Log Service project only if the Kubernetes clusters reside in the same region.

· How do I view the logs of Logtail?

The logs of Logtail are stored in the files named ilogtail.LOG and logtail_plugin.LOG in the /usr/local/ilogtail/ directory of a Logtail container. The stdout and stderr of the Logtail container are not for reference. You can ignore the following stdout and stderr:

start umount useless mount points, /shm\$|/merged\$|/mqueue\$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fbe2bdb95d13b1e110172ef57fe840c82155/merged: must be superuser to
umount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44beab6e69718/merged: must be superuser to
umount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640b1e16c22dbe/merged: must be superuser to
umount
......
xaros: umount: exited with status 255; aborting

xargs: umount: exited with status 255; aborting umount done start logtail ilogtail is running logtail status: ilogtail is running

· How do I view the status of Log Service components in Kubernetes clusters?

Run the following commands:

kubectl get deploy alibaba-log-controller -n kube-system kubectl get ds logtail-ds -n kube-system

- What do I do if alibaba-log-controller fails to start?
 - Check whether alibaba-log-controller is installed by using the following method:
- Run the installation command on the control plane of your Kubernetes cluster.
- Specify the ID of your Kubernetes cluster in the installation command.

If alibaba-log-controller is not installed by using the preceding method, run the kubect1 delete -f deploy command to delete the installation template that is generated. Then, run the installation command again.

• How do I view the status of the Logtail DaemonSet in a Kubernetes cluster?

Run the kubectl get ds -n kube-system command to view the status of the Logtail DaemonSet.

? Note

The default namespace to which a Logtail container belongs is kube-system.

• How do I view the version number, IP address, startup time, and status of Logtail?

• Run the following command to view the status of Logtail:

kubectl get po -n kube-system | grep logtail

The following output is returned:

NAME READY STATUS RESTARTS AGE logtail-ds-gb92k 1/1 Running 0 2h logtail-ds-wm71w 1/1 Running 0 4d

$\circ~$ Run the following command to view the version number and IP address of Logtail:

kubectl exec logtail-ds-gb92k -n kube-system cat /usr/local/ilogtail/app_info.json

The following output is returned:

```
{
   "UUID": "",
   "hostname": "logtail-ds-gb92k",
   "instance_id": "0EBB2B0E-0A3B-11E8-B0CE-0A58AC140402_172.20.4.2_1517810940",
   "ip": "192.0.2.0",
   "logtail_version": "0.16.2",
   "logtail_version": "0.16.2",
   "os": "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
   "update_time": "2021-02-05 06:09:01"
}
```

• How do I view the run logs of Logtail?

The run logs of Logtail are stored in the ilogtail.LOG file in the /usr/local/ilogtail/ directory. If the log file is rotated, the generated files are compressed and stored as ilogtail.LOG.x.gz.

Run the kubectl exec logtail-ds-gb92k -n kube-system tail /usr/local/ilogtail/ilogtail.LOG command to view the logs. Example output:

[2018-02-05 06:09:02.168693] [INF0] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:104] logtail plugin Resume:start [2018-02-05 06:09:02.168807] [INF0] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:106] logtail plugin Resume:success [2018-02-05 06:09:02.168822] [INF0] [9] [build/release64/sls/ilogtail/EventDispatcher.cpp:369] start add existed check point events, size:0 [2018-02-05 06:09:02.168827] [INF0] [9] [build/release64/sls/ilogtail/EventDispatcher.cpp:511] add existed check point events, size:0 cache size:0 event size:0 success count:0

· How do I restart Logtail for a pod?

i. Stop Logtail.

In the following command, logtail-ds-gb92k -n specifies the container, and kube-system specifies the namespace. Configure the parameters based on your business requirements.

kubectl exec logtail-ds-gb92k -n kube-system /etc/init.d/ilogtaild stop

If the following output is returned, Logtail is stopped:

```
kill process Name: ilogtail pid: 7
kill process Name: ilogtail pid: 9
stop success
```

ii. Start Logtail.

In the following command, logtail-ds-gb92k -n specifies the container, and kube-system specifies the namespace. Configure the parameters based on your business requirements.

kubectl exec logtail-ds-gb92k -n kube-system /etc/init.d/ilogtaild start

If the following output is returned, Logtail is started:

ilogtail is running

What to do next

Create Logtail configurations to collect container logs.

- DaemonSet mode
 - For more information about how to collect container logs by using CRDs, see Use CRDs to collect container logs in DaemonSet mode.
 - For more information about how to collect container stdout and stderr by using the Log Service console, see Use the Log Service console to collect container stdout and stderr in DaemonSet mode.
 - For more information about how to collect container text logs by using the Log Service console, see Use the Log Service console to collect container text logs in DaemonSet mode.
- Sidecar mode
 - For more information about how to collect container text logs by using CRDs, see Use CRDs to collect container text logs in Sidecar mode.
 - For more information about how to collect container text logs by using the Log Service console, see Use the Log Service console to collect container text logs in Sidecar mode.

4.3.1.5.3. Use the Log Service console to collect container text logs in DaemonSet

mode

This topic describes how to create a Logtail configuration in the Log Service console and use the Logtail configuration to collect container text logs in DaemonSet mode.

Prerequisites

The Logtail component is installed. For more information, see Install the Logtail component.

Features

Logtail can collect container text logs, and then upload the text logs together with container metadata to Log Service. Logtail supports the following features:

- Allows you to specify a log file path in a container. You do not need to manually map the log file path to a path on the host.
- Uses the container label whitelist to specify containers from which text logs are collected.
- Uses the container label blacklist to specify containers from which text logs are not collected.
- Uses the environment variable whitelist to specify containers from which text logs are collected.
- Uses the environment variable blacklist to specify containers from which text logs are not collected.
- Collects multi-line logs. For example, Logtail can collect Java stack logs.
- Automatically associates container metadata that needs to be uploaded together with the collected container text logs. The metadata includes container names, image names, pod names, namespaces, and environment variables.
- If a container runs in a Kubernetes cluster, Logtail also supports the following features:
- Uses Kubernetes namespaces, pod names, and container names to specify containers from which text logs are collected.
- $\circ~$ Uses the Kubernetes label whitelist to specify containers from which text logs are collected.
- Uses the Kubernetes label blacklist to specify containers from which text logs are not collected.
- Automatically associates Kubernetes labels that need to be uploaded together with the collected container text logs.

Limits

- If Logtail detects the die event on a container that is stopped, Logtail no longer collects text logs from the container. If collection latency exists, some text logs that are collected before the container is stopped may be lost.
- For Docker containers, only overlay and overlay2 storage drivers are supported. If other storage drivers are used, you must mount a volume to the directory of logs. Then, a temporary directory is generated.
- Logtail cannot access the symbolic link of a container. You must specify an actual path as the collection directory.

- If a volume is mounted to the data directory of a container, Logtail cannot collect data from the parent directory of the data directory. You must specify the complete path of the data directory as the collection directory.
- For example, if a volume is mounted to the /var/log/service directory and you set the collection directory to /var/log, Logtail cannot collect logs from the /var/log directory. You must specify /var/log/service as the collection directory.
- By default, Kubernetes mounts the root directory of the host to the /logtail_host directory of the Logtail container. If you want to collect text logs from the host, you must specify /logtail_host as the prefix of the log file path.
- For example, if you want to collect logs from the /home/logs/app_log/ directory of the host, you must specify /logtail_host/home/logs/app_log/ as the log file path.
- Logtail collects data from containers that use the Docker engine or containerd engine.
- Docker: Logtail accesses the Docker engine in the /run/docker.sock directory. Make sure that the directory exists and Logtail has the permissions to
 access the directory.
- containerd: Logtail accesses the containerd engine in the /run/containerd/containerd.sock directory. Make sure that the directory exists and Logtail
 has the permissions to access the directory.

Create a Logtail configuration

- 1. Log on to the Log Service console
- 2. In the Import Data section, click Kubernetes Object.
- 3. Select a project and a Logstore. Then, click **Next**.
- In this example, select the project that you use to install the Logtail component and the Logstore that you create.
- Click Use Existing Machine Groups. After you install the Logtail component, Log Service automatically creates a machine group named k8s-group-\${your_k8s_cluster_id}. You can select this machine group.
- 5. Select the k8s=group-\${your_k8s_cluster_id} machine group from Source Server Groups and move the machine group to Applied Server Groups. Then, click Next.

() Important

If the heartbeat status of the machine group is FAIL, you can click Automatic Retry. If the issue persists, see What do I do if no heartbeat connections are detected on Logtail? in the FAQ.

6. Configure the parameters for the Logtail configuration and click Next.

- i. Configure the basic settings, such as the name, log path, and mode. For more information, see Collect text logs.
- ii. Turn on Docker File.
- iii. Optional:Specify conditions to filter containers.
 - For versions earlier than Logtail V1.0.29, containers can be filtered only by using environment variables and container labels. The following table describes the parameters.

A namespace of a Kubernetes cluster and the name of a container in a Kubernetes cluster can be mapped to container labels. The value of the LabelKey parameter for a namespace is io.kubernetes.pod.namespace. The value of the LabelKey parameter for a container name is io.kubernetes.container.name . We recommend that you use the two container labels to filter containers. If the container labels do not meet your business requirements, you can use the environment variable whitelist or the environment variable blacklist to filter containers. For example, the namespace of a pod is backend-prod, and the name of a container in the pod is worker-server. If you want the logs of the worker-server container to be collected, you can specify io.kubernetes.pod.namespace : backend-prod or io.kubernetes.container.name : worker-server in the container label whitelist.

- () Important
 - Container labels are retrieved by running the docker inspect command. Container labels are different from Kubernetes labels. For more information, see Obtain container labels.
 - Environment variables are the same as the environment variables that are configured to start containers. For more information, see Obtain container environment variables.
 - Do not specify duplicate values for the LabelKey parameter. If you specify duplicate values for the LabelKey parameter, only one of the values takes effect.

Parameter	Description
Label Whitelist	 The container label whitelist. The whitelist specifies the containers from which text logs are collected. When you configure the container label whitelist, the LabelKey parameter is required, and the LabelValue parameter is optional. If the LabelValue parameter is empty, containers whose container labels contain the keys specified by LabelKey are matched. If the LabelValue parameter is not empty, containers whose container labels consist of the key-value pairs specified by LabelKey and LabelValue parameter is not empty, containers whose container labels consist of the key-value pairs specified by LabelKey and LabelValue parameter is not empty, containers whose container labels consist of the key-value pairs specified by LabelKey and LabelValue are matched. By default, string matching is performed for the values of the LabelValue parameter. Containers are matched only if the values of the container labels are the same as the values of the LabelValue parameter. If you specify a value that starts with a caret (^) and ends with a dollar sign (\$) for the LabelValue parameter, regular expression matching is performed. For example, if you set the LabelKey parameter to <code>io.kubernetes.container.name</code> and set the LabelValue parameter to <code>inginx(cube)\$</code>, a container named nginx and a container named cube are matched. Key-value pairs are connected by using the OR operator. If a container label consists of one of the specified key-value pairs, the container to which the container label belongs is matched.

Label Blacklist	 The container label blacklist. The blacklist specifies the containers from which text logs are not collected. When you configure the container label blacklist, the LabelKey parameter is required, and the LabelValue parameter is optional. If the LabelValue parameter is empty, containers whose container labels contain the keys specified by LabelKey are filtered out. If the LabelValue parameter is not empty, containers whose container labels consist of the key-value pairs specified by LabelKey and LabelValue parameter is not empty, containers whose container labels consist of the key-value pairs specified by LabelKey and LabelValue are filtered out. By default, string matching is performed for the values of the LabelValue parameter. Containers are matched only if the values of the container labels are the same as the values of the LabelValue parameter. If you specify a value that starts with a caret (^) and ends with a dollar sign (s) for the LabelValue parameter, regular expression matching is performed. For example, if you set the LabelKey parameter to <code>io.kubernetes.container.name</code> and set the LabelValue parameter to <code>io.ginx.cube)s</code>, a container named nginx and a container named cube are matched. Key-value pairs are connected by using the OR operator. If a container label consists of one of the specified key-value pairs, the container to which the container label belongs is filtered out.
Environment Variable Whitelist	The environment variable whitelist. The whitelist specifies the containers from which text logs are collected. When you configure the environment variable whitelist, the EnvKey parameter is required, and the EnvValue parameter is optional. If the EnvValue parameter is empty, containers whose environment variables contain the keys specified by EnvKey are matched. If the EnvValue parameter is not empty, containers whose environment variables consist of the key-value pairs specified by EnvKey are matched. By default, string matching is performed for the values of the EnvValue parameter. Containers are matched only if the values of the environment variables are the same as the values of the EnvValue parameter. If you specify a value that starts with a caret () and ends with a dollar sign (\$) for the EnvValue parameter, regular expression matching is performed. For example, if you set the EnvKey parameter to real starts whose port number is 80 and containers whose port number is 6379 are matched. Key-value pairs are connected by using the OR operator. If an environment variable consists of one of the specified key-value pairs, the container to which the environment variable belongs is matched.
Environment Variable Blacklist	 The environment variable blacklist. The blacklist specifies the containers from which text logs are not collected. When you configure the environment variable blacklist, the EnvKey parameter is required, and the EnvValue parameter is optional. If the EnvValue parameter is empty, containers whose environment variables contain the keys specified by EnvKey are filtered out. If the EnvValue parameter is not empty, containers whose environment variables consist of the key-value pairs specified by EnvKey and EnvValue are filtered out. By default, string matching is performed for the values of the EnvValue parameter. Containers are matched only if the values of the environment variables are the same as the values of the EnvValue parameter. If you specify a value that starts with a caret (^) and ends with a dollar sign (\$) for the EnvValue parameter, regular expression matching is performed. For example, if you set the EnvKey parameter to matched only is 80 and containers whose port number is 6379 are matched. Key-value pairs are connected by using the OR operator. If an environment variable consists of one of the specified key-value pairs, the container to which the environment variable belongs is filtered out.

For Logtail V1.0.29 or later, we recommend that you use different levels of Kubernetes information, such as pod names, namespaces, container names, and labels to filter containers.

Turn on **Deployed in K8s** and configure the following parameters to filter containers.

? Note

If you change Kubernetes labels when Kubernetes control resources, such as Deployments, are running, the operational pod is not restarted. Therefore, the pod cannot detect the change. This may cause a matching rule to become invalid. When you specify the Kubernetes label whitelist and the Kubernetes label blacklist, we recommend that you use the Kubernetes labels of pods.

Parameter	Description
K8s Pod Name Regular Matching	The pod name. The pod name specifies the containers from which text logs are collected. Regular expression matching is supported. For example, if you specify (nginw-log-demo.*/), all containers in the pod whose name starts with nginx-log-demo are matched.
K8s Namespace Regular Matching	The namespace. The namespace specifies the containers from which text logs are collected. Regular expression matching is supported. For example, if you specify <pre>^(default nginx)\$</pre> , all containers in the nginx and default namespaces are matched.
K8s Container Name Regular Matching	The container name. The container name specifies the containers from which text logs are collected. Regular expression matching is supported. Kubernetes container names are defined in spec.containers. For example, if you specify <u>^(container-test)</u> , all containers whose name is container-test are matched.
K8s Label Whitelist	 The Kubernetes label whitelist. The whitelist specifies the containers from which text logs are collected. When you configure the Kubernetes label whitelist, the LabelKey parameter is required, and the LabelValue parameter is optional. If the LabelValue parameter is empty, containers whose Kubernetes labels contain the keys specified by LabelKey are matched. If the LabelValue parameter is not empty, containers whose Kubernetes labels consist of the key-value pairs specified by LabelKey and LabelValue are matched. By default, string matching is performed for the values of the LabelValue parameter. Containers are matched only if the values of the Kubernetes labels are the same as the value of the LabelValue parameter. If you specify a value that starts with a caret (^) and ends with a dollar sign (s), regular expression matching is performed. For example, if you set the LabelKey parameter to app and set the LabelValue parameter to <u>`(teetlitest2)</u>, containers whose Kubernetes labels consist of app:test1 or app:test2 are matched. Key-value pairs are connected by using the OR operator. If a Kubernetes label consists of one of the specified keyvalue pairs, the container to which the Kubernetes label belongs is matched.
K8s Label Blacklist	 The Kubernetes label blacklist. The blacklist specifies the containers from which text logs are not collected. When you configure the Kubernetes label blacklist, the LabelKey parameter is required, and the LabelValue parameter is optional. If the LabelValue parameter is empty, containers whose Kubernetes labels contain the keys specified by LabelKey are filtered out. If the LabelValue parameter is not empty, containers whose Kubernetes labels consist of the key-value pairs specified by LabelKey and LabelValue are filtered out. By default, string matching is performed for the values of the LabelValue parameter. Containers are matched only if the values of the Kubernetes labels are the same as the values of the LabelValue parameter. If you specify a value that starts with a caret (^) and ends with a dollar sign (\$), regular expression matching is performed. Key-value pairs are connected by using the OR operator. If a Kubernetes label consists of one of the specified key-value pairs, the container to which the Kubernetes label belongs is filtered out.

iv. Optional:Specify log labels.

For Logtail V1.0.29 or later, we recommend that you specify environment variables and Kubernetes labels for logs as log labels.

Parameter	Description
Environment Variable Log Tag	After you specify environment variables as log labels, Log Service adds environment variable-related fields to logs. For example, if you set the EnvKey parameter to vERSION and set the EnvValue parameter to env_version, Log Service adds the tag : env version_: v1.0.0 field to logs if the environment variable configurations of a container include VERSION=v1.0.0.
K8s Label Log Tag	After you specify Kubernetes labels as log labels, Log Service adds Kubernetes label-related fields to logs. For example, if you set the LabelKey parameter to app and set the LabelValue parameter to k8s_label_app , Log Service adds the taq : k8s label_app_: serviceA field to logs if the label configurations of a Kubernetes cluster include app=serviceA .

7. Preview data, configure indexes, and then click Next.

By default, full-text indexing is enabled for Log Service. You can also configure field indexes based on collected logs in manual or automatic mode. For more information, see Configure indexes.

Configuration examples

Example 1: Filter containers based on the environment variable whitelist and the environment variable blacklist

Collect text logs from the containers whose environment variable configurations include NGINX_SERVICE_PORT=80 but exclude POD_NAMESPACE=kube-system . The log file path is /var/log/nginx/access.log . The logs are parsed in simple mode.

1. Obtain environment variables.

To view the environment variables of a container, you can log on to the host on which the container resides.



2. Create a Logtail configuration.

The following figure shows an example of a Logtail configuration. For more information about how to create a Logtail configuration that is used to collect logs in simple mode, see Collect logs in simple mode.

* Config Name:	docker-file			
	Import Other Configuration			
* Log Path:	/var/log/nginx	/**/	access.log	
	All files under the specified folder (including all of be monitored. The file name can be a complete must start with "/"; for example, /apsara/nuwa/ example, C:\Program Files\Intel*.Log.	lirectory leve name or a n /app.Log. Th	els) that conform to the file name ame that contains wildcards. The ne Windows file path must start w	convention will Linux file path vith a drive; for
Blacklist:				
	You can configure a blacklist to skip the specifie the specified directories and files support exact <i>Implmydir</i> directory as a filtering condition, you <i>Implmydir</i> /file directory as a filtering condition, y directory. Documentation	d directories match and w can skip all t rou can skip	or files during log data collection vildcard match. For example, if yo files in the directory. If you specifi only the specified file in the	n. The names of ou specify the y the
Docker File:				
	For a Docker file, you can directly configure the the configuration of the label whitelist and black will automatically monitor the creation and destr containers according to the specified tags. For n	log path and ist and envir uction of cor nore informa	I container tags. Container tags a onment variable whitelist and bla tainers, and collect log entries of tion, see Documentation	are specified by acklist. Logtail f the specified
Label Whitelist:	LabelKey 🕂	LabelN	/alue	Delete
	Collect the logs from Docker container in the wh	itelist (empt	/ means collect all logs)	
Label Blacklist:	LabelKey 🕂	LabelV	/alue	Delete
	Do not collect logs from Docker containers in the	e blacklist (e	mpty means collecting all logs)	
Environment Variable	EnvKey +	EnvValue		Delete
Whitelist:	NGINX_PORT_80_TCP_PORT	80		×
	Collects log entries that contain the environmen entries will be collected.	t variables in	the whitelist. If the whitelist is er	npty, all log
Environment Variable	EnvKey 🕂	EnvValue		Delete
Blacklist:	POD_NAMESPACE	kube-syst	em	×

Example 2: Filter containers based on the container label whitelist and the container label blacklist

Collect text logs from the containers whose container label is io.kubernetes.container.name=nginx . The log file path is /var/log/nginx/access.log . The logs are parsed in **simple mode**.

1. Obtain container labels.

To view the container labels of a container, you can log on to the host on which the container resides.

"OnBuild": null,		
"Labels": {		
"annotation.io.kubernetes.co	ntainer.hash": "53073f5a",	
"annotation.io.kubernetes.co	ntainer.restartCount": "0",	
"annotation.io.kubernetes.co	ntainer.terminationMessagePath": "/dev/termination-log",	
"annotation.io.kubernetes.co	ontainer.terminationMessagePolicy": "File",	
"annotation.io.kubernetes.pd	od.terminationGracePeriod": "30",	
"io.kubernetes.container.log	path": "/var/log/pods/ad00a078-4182 585/nginx_0.log",	
"io.kubernetes.container.nam	ne": "nginx",	
"10.kubernetes.docker.type":	"container",	
"io.kubernetes.pod.name": "e	example-foo-86ccd54874-r4mfh",	
"io.kubernetes.pod.namespace	:": "default",	
"io.kubernetes.pod.uid": "ad		
"io.kubernetes.sandbox.id":	"Site reserves. of issuessably a cash site assues according 1dfa6da112969",	
"maintainer": "NGINX Docker	Maintainers <docker-maint@nginx.com>"</docker-maint@nginx.com>	
},		
"StopSignal": "SIGTERM"		
2. Create a Logtail configuration.		

The following figure shows an example of a Logtail configuration. For more information about how to create a Logtail configuration that is used to collect logs in simple mode, see Collect logs in simple mode.

	Import Other Configuration			
* Log Path:	/var/log/nginx	/**/	access.log	
	All files under the specified folder (including a be monitored. The file name can be a comple must start with "/"; for example, /apsara/nuwa example, C:\Program Files\Intel*.Log.	II directory leve te name or a r I/app.Log. Ti	els) that conform to the file name that contains wildcard he Windows file path must s	name convention will s. The Linux file path start with a drive; for
Blacklist:				
	You can configure a blacklist to skip the speci the specified directories and files support exa /tmp/mydir directory as a filtering condition, yr /tmp/mydirfile directory as a filtering condition directory. Documentation	ified directorie: ct match and v ou can skip all n, you can skip	s or files during log data col wildcard match. For example files in the directory. If you a o only the specified file in the	lection. The names o e, if you specify the specify the e
Docker File:				
	For a Docker file, you can directly configure the configuration of the label whitelist and bla will automatically monitor the creation and de containers according to the specified tags. For	he log path and cklist and envi struction of co or more informa	d container tags. Container ronment variable whitelist a ntainers, and collect log ent ation, see Documentation	tags are specified by nd blacklist. Logtail ries of the specified
Label Whitelist:	For a Docker file, you can directly configure II the configuration of the label whitelist and bla will automatically monitor the creation and de containers according to the specified tags. For LabelKey +	he log path and cklist and envi struction of co or more informa LabelValu	d container tags. Container ronment variable whitelist a ntainers, and collect log ent ation, see Documentation	tags are specified by nd blacklist. Logtail ries of the specified Delete
Label Whitelist:	For a Docker file, you can directly configure the configuration of the label whitelist and bla will automatically monitor the creation and de containers according to the specified tags. For LabelKey + LabelKey + Io kubmetes container name	he log path and cklist and envi struction of co or more informa Label/Valu	d container tags. Container romment variable whitelist a ntainers, and collect log ent ation, see Documentation	tags are specified by nd blacklist. Logtali ries of the specified Delete
Label Whitelist:	For a Docker file, you can directly configure th the configuration of the label whitelist and bla will automatically monitor the creation and de containers according to the specified tags. For LabelKey + io.kubmetes.container.name Collect the logs from Docker container in the	he log path and cklist and envi struction of coor r more informa LabelValu nginx whitelist (empt	d container tags. Container ronment variable whitelist a ntainers, and collect log ent ation, see Documentation ue	tags are specified by nd blacklist. Logtail rises of the specified Delete
Label Whitelist: Label Blacklist:	For a Docker file, you can directly configure the configuration of the label whitelist and bla will automatically monitor the creation and de containers according to the specified tags. For LabelKey + io kubmetes container name Collect the logs from Docker container in the LabelKey +	he log path and cklist and envi struction of co r more information LabelValu mginx LabelValu	d container tags. Container ronment variable whitelist a ntainers, and collect log ent ation, see Documentation ue ty means collect all logs) ue	tags are specified by nd blacklist. Logtail iries of the specified Delete Delete Delete
Label Whitelist Label Blacklist	For a Docker file, you can directly configure the configuration of the label whitelist and bla will automatically monitor the creation and de containers according to the specified tags. For LabelKey + io.kubmetes.container.name Collect the logs from Docker container in the LabelKey + type	he log path an cklist and envi struction of co r more information LabelValu nginx whitelist (empt LabelValu pre	d container tags. Container ronment variable whitelist a ntainers, and collect log ent ation, see Documentation ue by means collect all logs) ue	tags are specified by blacklist. Logtail rises of the specified Delete

Example 3: Filter containers by using Kubernetes namespaces, pod names, and container names

Collect text logs from the nginx-log-demo-0 container in pods whose name starts with nginx-log-demo in the default namespace. 1. Obtain different levels of Kubernetes information.

Obtain information about pods.

~/.kube » kubectl get po	ds			a a a
NAME	READY	STATUS	RESTARTS	AGE
nginx-log-demo-0-bxl79	1/1	Running	0	48d
nginx-log-demo-1-qmrqk	1/1	Running	0	48d
nginx-log-demo-2-7khv9	1/1	Running	0	48d
nginx-log-demo-3-j24xc	1/1	Running	0	48d

• Obtain information about namespaces.

apiVersion: v1
kind: Pod
metadata:
annotations:
kubernetes.io/psp: ack.privileged
creationTimestamp: "2022-01-06T18:42:43Z"
generateName: nainx-log-demo-0-
labels:
controller-uid: ge3eedc4-1667-458b-g6fe-39888576dbf4
inh-name, nainx-log-damo-0
name: nainy-log-dem-0-by179
namenace default
namespace. default
- millerices hatch/ul
blockOurseDation: true
southeller two
controller: true
kind: Joo
name: nginx-log-aemo-0
uld: deseedc4-lbb/-458b-dbte-398885/6dbt4
resourceVersion: "50566856"
uid: ee10fb7d-d989-47b3-bc2a-e9ffbe767849
spec:
containers:
- args:
log-type=nginx
stdout=true
total-count=10000000000
log-file-size=1000000000
log-file-count=2
logs-per-sec=2778
command:
- /bin/mock_log
image: registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest
imagePullPolicy Always
name: nainx-log-demo-0
Pasoupcas:

2. Create a Logtail configuration.

For more information about how to create a Logtail configuration that is used to collect logs in simple mode, see Collect logs in simple mode.

Example 4: Filter containers by using Kubernetes labels

Collect text logs from containers whose Kubernetes labels contain the job-name key and a specific value. The value starts with nginx-log-demo.

1. Obtain Kubernetes labels.

apiVersion: v1
kind: Pod
metadata:
annotations:
kubernetes.io/psp: ack.privileged
creationTimestamp: "2022-01-06T18:42:43Z"
generateName: nginx-log-demo-0-
labels:
controller-uid: ae3eedc4-1667-458b-a6fe-39888576dbf4
job-name: nginx-log-demo-0
name: nginx-log-demo-0-bx179
namespace: default
ownerReferences:
- apiVersion: batch/v1
blockOwnerDeletion: true
controller: true
kind: Job
name: nginx-log-demo-0
uid: ae3eedc4-1667-458b-a6fe-39888576dbf4
resourceVersion: "50566856"
uid: ee10fb7d-d989-47b3-bc2a-e9ffbe767849

2. Create a Logtail configuration.

For more information about how to create a Logtail configuration that is used to collect logs in simple mode, see Collect logs in simple mode.

Default fields

The following table describes the fields that are included by default in each container text log.

Log field	Description
_image_name_	The name of the image.
_container_name_	The name of the container.
_pod_name_	The name of the pod.
namespace	The namespace of the pod.

_pod_uid_	The unique identifier of the pod.
_container_ip_	The IP address of the pod.

4.3.1.5.4. Use the Log Service console to collect container stdout and stderr in

DaemonSet mode

This topic describes how to create a Logtail configuration in the Log Service console and use the Logtail configuration to collect container stdout and stderr in DaemonSet mode.

Prerequisites

- The Logtail component is installed. For more information, see Install the Logtail component.
- A Logstore is created in the project that you use to install the Logtail component. For more information, see Create a Logstore.

Features

Logtail can collect container stdout and stderr, and then upload the stdout and stderr together with container metadata to Log Service. Logtail supports the following features:

- Collects stdout and stderr.
- Uses the container label whitelist to specify containers from which stdout and stderr are collected.
- Uses the container label blacklist to specify containers from which stdout and stderr are not collected.
- Uses the environment variable whitelist to specify containers from which stdout and stderr are collected.
- Uses the environment variable blacklist to specify containers from which stdout and stderr are not collected.
- Collects multi-line logs. For example, Logtail can collect Java stack logs.
- Automatically associates container metadata that needs to be uploaded together with the collected container stdout and stderr. The metadata
- includes container names, image names, pod names, namespaces, and environment variables.
- If a container runs in a Kubernetes cluster, Logtail also supports the following features:
 - Uses Kubernetes namespaces, pod names, and container names to specify containers from which stdout and stderr are collected.
- $\circ~$ Uses the Kubernetes label whitelist to specify containers from which stdout and stderr are collected.
- Uses the Kubernetes label blacklist to specify containers from which stdout and stderr are not collected.
- Automatically associates Kubernetes labels that need to be uploaded together with the collected container stdout and stderr.

Implementation

Logtail communicates with the domain socket of Docker. Logtail queries all Docker containers and identifies the containers from which stdout and stderr are collected by using the specified labels and environment variables. Logtail runs the docker logs command to collect logs from the specified containers.

When Logtail collects stdout and stderr from a container, Logtail periodically stores checkpoints to a checkpoint file. If Logtail is stopped and then started, Logtail collects logs from the last checkpoint.

Limits

- You can use the Log Service console to collect stdout and stderr in DaemonSet mode only if Logtail runs V0.16.0 or later and runs on Linux. For more information about Logtail versions and version updates, see Install Logtail on a Linux server.
- Logtail collects data from containers that use the Docker engine or containerd engine.
- Docker: Logtail accesses the Docker engine in the /run/docker.sock directory. Make sure that the directory exists and Logtail has the permissions to
 access the directory.
- containerd: Logtail accesses the containerd engine in the /run/containerd/containerd.sock directory. Make sure that the directory exists and Logtail
 has the permissions to access the directory.
- By default, the last multi-line log that is collected by Logtail is cached for 3 seconds. This prevents the multi-line log from being split into multiple logs due to output latency. You can change the cache time by modifying the BeginLineTimeoutMs parameter. We recommend that you do not specify a value less than 1000 with millisecond precision. If you specify a value that is less than 1000, an error may occur.
- If Logtail detects the die event on a container that is stopped, Logtail no longer collects stdout or stderr from the container. If collection latency exists, some stdout and stderr that are collected before the container is stopped may be lost.
- The logging driver collects stdout and stderr only in the JSON format from containers that use the Docker engine.
- By default, stdout and stderr that are collected from different containers by using the same Logtail configuration have the same context. If you want to specify a different context for the stdout and stderr that are collected from each container, you must create a Logtail configuration for each container.
- By default, the collected data is stored in the content field. Logtail can process the collected data. For more information, see Customize Logtail plug-ins to process data.

Create a Logtail configuration

- 1. Log on to the Log Service console
- 2. In the Import Data section, click Kubernetes Standard Output.
- 3. Select a project and a Logstore. Then, click Next.

In this example, select the project that you use to install the Logtail component and the Logstore that you create.

- 4. Click Use Existing Machine Groups.
 After you install the Logtail component, Log Service automatically creates a machine group named k8s-group-\${your_k8s_cluster_id}. You can select this machine group.
- 5. Select the k8s-group-\${your_k8s_cluster_id} machine group from Source Server Groups and move the machine group to Applied Server Groups. Then, click Next.

() Important

If the heartbeat status of the machine group is FAIL, you can click Automatic Retry. If the issue persists, see What do I do if no heartbeat connections are detected on Logtail? in the FAQ.

6. In the Specify Data Source step, specify the data source and click Next.

Configure the parameters that are used to collect logs in the Plug-in Config field. Example:

1
"inputs":[
{
"type":"service_docker_stdout",
"detail":{
"Stdout":true,
"Stderr":true,
"IncludeContainerLabel":{
"LabelKey":"LabelValue"
},
"ExcludeContainerLabel":{
"LabelKey":"LabelValue"
},
"IncludeK8sLabel":{
"LabelKey":"LabelValue"
},
"ExcludeK8sLabel":{
"LabelKey":"LabelValue"
},
"IncludeEnv":{
"EnvKey":"EnvValue"
},
"ExcludeEnv":{
"EnvKey":"EnvValue"
},
"ExternalK8sLabelTag":{
"EnvKey":"EnvValue"
},
"ExternalEnvTag":{
"EnvKey":"EnvValue"
},
"K8sNamespaceRegex":"^(default kube-system)\$",
"K8sPodRegex":"^(deploy.*)\$",
"K8sContainerRegex":"^(container1 container2)\$
}
}
]
}

Configure the following parameters:

Data source type

The type of the data source is fixed as service_docker_stdout.

• Parameters related to container filtering

• For versions earlier than Logtail V1.0.29, you can filter containers only by using environment variables or container labels. You can take note of the following descriptions.

The namespace of a Kubernetes cluster and the name of a container in the Kubernetes cluster can be mapped to container labels. The value of the LabelKey parameter for the namespace is io.kubernetes.pod.namespace. The value of the LabelKey parameter for the container name is io.kubernetes.container.name we recommend that you use the two container labels to filter containers. If the container labels do not meet your business requirements, you can use the environment variable whitelist or the environment variable blacklist to filter containers. For example, the namespace of a pod is backend-prod, and the name of a container in the pod is worker-server. If you want the logs of the worker-server" in the container label whitelist.

() Important

- Container labels are retrieved by running the docker inspect command. Container labels are different from Kubernetes labels. For more information, see Obtain container labels.
- Environment variables are the same as the environment variables that are configured to start containers. For more information, see Obtain environment variables.
- Do not specify duplicate values for the LabelKey parameter. If you specify duplicate values for the LabelKey parameter, only one of the values takes effect.

Parameter	Туре	Required	Description
-----------	------	----------	-------------

User Guide-Log Service

IncludeLabel	Map (The values of the LabelKey and LabelValue parameters are strings.)	No	 The container label whitelist. The whitelist specifies the containers from which stdout and stderr are collected. This parameter is empty by default, which indicates that stdout and stderr are collected from all containers. When you configure the container label whitelist, the LabelKey parameter is required, and the LabelValue parameter is optional. If the LabelValue parameter is empty, containers whose container labels contain the keys specified by LabelKey are matched. If the LabelValue parameter is not empty, containers whose container labels consist of the key-value parameter is not empty, containers whose container labels consist of the key-value parameter is not empty, containers whose container labels consist of the key-value parameter is proformed for the values of the LabelValue parameter. By default, string matching is performed for the values of the container labels are the same as the values of the LabelValue parameter. If you specify a value that starts with a cort (~) and ends with a dollar sign (\$) for the LabelValue parameter to [o.kubernetes.container.name] and set the LabelValue parameter to [o.kubernetes.container named nginx and a container named cube are matched. Key-value pairs are connected by using the OR operator. If a container label consists of one of the specified key-value pairs, the container to which the container label belongs is matched.
ExcludeLabel	Map (The values of the LabelKey and LabelValue parameters are strings.)	No	 The container label blacklist. The blacklist specifies the containers from which stdout and stderr are not collected. This parameter is empty by default, which indicates that stdout and stderr are collected from all containers. When you configure the container label blacklist, the LabelKey parameter is required, and the LabelValue parameter is optional. If the LabelValue parameter is empty, containers whose container labels contain the keys specified by LabelKey are filtered out. If the LabelValue parameter is not empty, containers whose container labels consist of the key-value pairs specified by LabelKey and LabelValue are filtered out. By default, string matching is performed for the values of the container labels are the same as the values of the LabelValue parameter. If you specify a value that starts with a caret (^) and ends with a dollar sign (\$) for the LabelValue parameter, regular expression matching is performed. For example, if you set the LabelKey parameter to <u>io.kubernetes.container.name</u> and set the LabelValue parameter to <u>(regine(cube)\$</u>, a container named nginx and a container named cube are matched. Key-value pairs are connected by using the OR operator. If a container label consists of one of the specified key-value pairs, the container to which the container label belongs is filtered out.
IncludeEnv	Map (The values of the EnvKey and EnvValue parameters are strings.)	No	 The environment variable whitelist. The whitelist specifies the containers from which stdout and stderr are collected. This parameter is empty by default, which indicates that stdout and stderr are collected from all containers. When you configure the environment variable whitelist, the EnvKey parameter is required, and the EnvValue parameter is optional. If the EnvValue parameter is empty, containers whose environment variables contain the keys specified by EnvKey are matched. If the EnvValue parameter is not empty, containers whose environment variables consist of the key-value pairs specified by EnvKey and EnvValue are matched. By default, string matching is performed for the values of the EnvValue parameter. Containers are matched only if the values of the environment variables are the same as the values of the EnvValue parameter. If you specify a value that starts with a caret (^) and ends with a dollar sign (\$) for the EnvValue parameter, regular expression matching is performed. For example, if you set the EnvKey parameter to science.powr and set the EnvValue parameter to science.powr and

parameter to NOTING_SERVICE_PORT and set the EnvValue parameter to (0006379)\$, containers whose port number is 80 and containers whose port number is 6379 are matched. Key-value pairs are connected by using the OR operator. If an environment variable consists of one of the specified key-value pairs, the container to which the environment variable belongs is filtered out.	ExcludeEnv	Map (The values of the EnvKey and EnvValue parameters are strings.)	No	The environment variable blacklist. The blacklist specifies the containers from which stdout and stderr are not collected. This parameter is empty by default, which indicates that stdout and stderr are collected from all containers. When you configure the environment variable blacklist, the EnvKey parameter is required, and the EnvValue parameter is optional. If the EnvValue parameter is empty, containers whose environment variables contain the keys specified by EnvKey are filtered out. If the EnvValue parameter is not empty, containers whose environment variables consist of the key-value pairs specified by EnvKey and EnvValue are filtered out. By default, string matching is performed for the values of the EnvValue parameter. Containers are matched only if the values of the environment variables are the same as the values of the EnvValue parameter. If you specify a value that starts with a care(^) and ends with a dollar sign (\$) for the EnvValue parameter, regular expression matching is performed. For example, if you set the EnvKey parameter to NEDER SERVICE FORT and set the EnvValue parameter to NEDER SERVICE FORT and set the EnvValue parameter to NEDER SERVICE FORT and set the EnvValue parameter to Secondarys, containers whose port number is 80 and containers whose port number is 6379 are matched. Key-value pairs are connected by using the OR operator. If an environment variable consists of one of the specified key-value pairs, the container to which the anvironment variable paramet.
consists of one of the specified key-value pairs, the container to which the environment variable belongs is filtered out.				consists of one of the specified key-value pairs, the container to which the environment variable belongs is filtered out.

For Logtail V1.0.29 or later, we recommend that you use different levels of Kubernetes information, such as pod names, namespaces, container names, and labels to filter containers.

? Note

If you change Kubernetes labels when Kubernetes control resources, such as Deployments, are running, the operational pod is not restarted. Therefore, the pod cannot detect the change. This may cause a matching rule to become invalid. When you configure the Kubernetes label whitelist or the Kubernetes label blacklist, we recommend that you use the Kubernetes labels of pods. For more information about Kubernetes labels, see Labels and Selectors.

Parameter	Туре	Required	Description
IncludeK8sLabel	Map (The values of the LabelKey and LabelValue parameters are strings.)	No	The Kubernetes label whitelist. The whitelist specifies the containers from which stdout and stderr are collected. When you configure the Kubernetes label whitelist, the LabelKey parameter is required, and the LabelValue parameter is optional. If the LabelValue parameter is empty, containers whose Kubernetes labels contain the keys specified by LabelKey are matched. If the LabelValue parameter is not empty, containers whose Kubernetes labels consist of the key-value pairs specified by LabelKey and LabelValue are matched. By default, string matching is performed for the values of the LabelValue parameter. Containers are matched only if the values of the LabelValue parameter. Containers are matched only if the values of the Kubernetes labels are the same as the values of the LabelValue parameter. If you specify a value that starts with a caret (^) and ends with a dollar sign (\$), regular expression matching is performed. For example, if you specify a cancer to app and set the LabelValue parameter to ^ (test1[test2]\$, containers whose Kubernetes labels consist of app:test1 or app:test2 are matched. Key-value pairs are connected by using the OR operator. If a Kubernetes label consists of one of the specified key-value pairs, the container to which the Kubernetes label belongs is matched.
ExcludeK8sLabel	Map (The values of the LabelKey and LabelValue parameters are strings.)	No	The Kubernetes label blacklist. The blacklist specifies the containers from which stdout and stderr are not collected. When you configure the Kubernetes label blacklist, the LabelKey parameter is required, and the LabelValue parameter is optional. If the LabelValue parameter is empty, containers whose Kubernetes labels contain the keys specified by LabelKey are filtered out. If the LabelValue parameter is not empty, containers whose Kubernetes labels consist of the key-value pairs specified by LabelKey and LabelValue are filtered out. By default, string matching is performed for the values of the LabelValue parameter. Containers are matched only if the values of the LabelValue parameter. If you specify a value that starts with a caret () and ends with a dollar sign (s), regular expression matching is performed. For example, if you set the LabelVp parameter to app and set the LabelValue parameter to app. (test11test2)s, containers whose Kubernetes labels consist of one of the specified key-value pairs, the container to which the Kubernetes label belongs is filtered out.
K8sNamespaceReg ex	string	No	The namespace. The namespace specifies the containers from which stdout and stderr are collected. Regular expression matching is supported. For example, if you specify "K8sNamespaceRegex":"^(default nginx)\$", all containers in the nginx and default namespaces are matched.
K8sPodRegex	string	No	The pod name. The pod name specifies the containers from which stdout and stderr are collected. Regular expression matching is supported. For example, if you specify "K8sPodRegex":"^(nginx-log-demo.*)\$", all containers in the pod whose name starts with nginx-log-demo are matched.
K8sContainerRege x	string	No	The container name. The container name specifies the containers from which stdout and stderr are collected. Regular expression matching is supported. Kubernetes container names are defined in spec.containers. For example, if you specify "K8sContainerRegex":"^(container-test)\$", all containers whose name is container-test are matched.

• Parameters related to log labels

For Logtail V1.0.29 or later, we recommend that you specify environment variables or Kubernetes labels for logs as log labels.

Parameter	Туре	Required	Description
ExternalEnvTag	Map (The values of the EnvKey and EnvValue parameters are strings.)	No	After you specify environment variables as log labels, Log Service adds environment variable-related fields to logs. For example, if you set the EnvKey parameter to version and set the EnvValue parameter to environment, Log Service adds the tag_:_env_version_:v1.0.0 field to logs if the environment variable configurations of a container include VERSION=v1.0.0.

ExternalK8sLabelT ag	Map (The values of the LabelKey and LabelValue parameters are strings.)	No	After you specify Kubernetes labels as log labels, Log Service adds Kubernetes label- related fields to logs. For example, if you set the LabelKey parameter to pp and set the LabelValue parameter to k8s_label_app , Log Service adds the _tag_:_k8s_label_app_: serviceA field to logs if the label configurations of a Kubernetes cluster include app=serviceA .
-------------------------	---	----	--

• Other parameters

Parameter	Туре	Required	Description
Stdout	boolean	No	Specifies whether to collect stdout. This parameter is empty by default, which indicates that stdout is collected.
Stderr	boolean	No	Specifies whether to collect stderr. This parameter is empty by default, which indicates that stderr is collected.
BeginLineRegex	string	No	The regular expression that is used to match the beginning of the first line of a log. This parameter is empty by default, which indicates that each line is regarded as a log. If the beginning of a line matches the specified regular expression, the line is regarded as the first line of a new log. If the beginning of a line does not match the specified regular expression, the line is regarded as a part of the last log.
BeginLineTimeout Ms	int	No	The timeout period for matching the beginning of the first line of a log based on the specified regular expression. This parameter is empty by default, which indicates that the timeout period is 3,000 milliseconds. If no new log is generated within 3,000 milliseconds, Logtail stops matching the beginning of the first line of a log and uploads the last log to Log Service.
BeginLineCheckLen gth	int	No	The size of the beginning of the first line of a log that matches the specified regular expression. This parameter is empty by default, which indicates that the size of the beginning of the first line of a log is 10,240 bytes. You can configure this parameter to check whether the beginning of the first line of a log matches the specified regular expression. We recommend that you configure this parameter to improve the match efficiency.
MaxLogSize	int	No	The maximum size of a log. This parameter is empty by default, which indicates that the maximum size of a log is 524,288 bytes. If the size of a log exceeds the value of this parameter, Logtail stops matching the beginning of the first line of a log and uploads the log to Log Service.
StartLogMaxOffset	int	No	The maximum size of historical data that can be traced the first time Logtail collects logs from a log file. Valid values: [131072,1048576]. Unit: bytes. This parameter is empty by default. In this case, the maximum size of historical data that can be traced is 131,072 bytes, equivalent to 128 KB.

7. Preview data, configure indexes, and then click **Next**.

By default, full-text indexing is enabled for Log Service. You can also configure field indexes based on collected logs in manual or automatic mode. For more information, see Configure indexes.

Examples of Logtail configurations for single-line logs

Example 1: Filter containers based on the environment variable whitelist and the environment variable blacklist

Collect stdout and stderr from the containers whose environment variable configurations include NGINX_SERVICE_PORT=80 but exclude POD_NAMESPACE=kube-system .

1. Obtain environment variables.

To view the environment variables of a container, you can log on to the host on which the container resides.

"StdinOnce": false,
"Env": [
"HTTP_SVC_SERVICE_PORT_HTTP=80",
"LOG4J_APPENDER_DEMO_SPRING_BOOT_SVC_PORT= :8080",
"LOG4J_APPENDER_DEMO_SPRING_BOOT_SVC_PORT_8080_TCP_PORT=8080",
"HTTP_SVC_PORT_80_TCP_ADDR= ",
"NGINX_PORT_80_TCP=tcp://
"NGINX_PORT_80_TCP_PROTO=tcp",
"LOG4J_APPENDER_DEMO_SPRING_BOOT_SVC_SERVICE_PORT=8080",
"KUBERNETES_SERVICE_HOST= .",
"HTTP_SVC_SERVICE_HOST= ",
"HTTP_SVC_PORT_80_TCP_PROTO=tcp",
"NGINX_PORT_80_TCP_ADDR=: ",
"LOG4J_APPENDER_DEMO_SPRING_BOOT_SVC_PORT_8080_TCP_PROTO=tcp",
"KUBERNETES_SERVICE_PORT_HTTPS=443",
"KUBERNETES_PORT=tcp:// :443",
"NGINX_PORT=tcp://
"HTTP_SVC_PORT=tcp:// ::80",
"HTTP_SVC_PORT_80_TCP_PORT=80",
<pre>"NGINX_SERVICE_PORT=80",</pre>
"KUBERNETES_PORI_443_ICP=tcp:// :443",
"KUBERNETES_PORT_443_TCP_PROTO=tcp",
"HTTP_SVC_SERVICE_PORT=80",
"KUBERNETES_PORT_443_TCP_ADDR=17 1",
"HTTP_SVC_PORT_80_TCP=tcp:// :80".

2. Create a Logtail configuration.

Example:

```
"inputs": [
       {
           "type": "service_docker_stdout",
           "detail": {
              "Stdout": true,
               "Stderr": true,
              "IncludeEnv": {
                 "NGINX_SERVICE_PORT": "80"
               }.
               "ExcludeEnv": {
                   "POD_NAMESPACE": "kube-system"
               }
          }
      }
   ]
}
```

Example 2: Filter containers based on the container label whitelist and the container label blacklist

 $\label{eq:container} Collect \ stdout \ and \ stderr \ from \ the \ containers \ whose \ container \ label \ is \ \ \ io. \ kubernetes. \ container. \ name=nginx \ .$

1. Obtain container labels.

To view the container labels of a container, you can log on to the host on which the container resides.



2. Create a Logtail configuration.

Example:
Example 3: Filter containers by using Kubernetes namespaces, pod names, and container names

Collect stdout and stderr from the nginx-log-demo-0 container in pods whose name starts with nginx-log-demo in the default namespace. 1. Obtain different levels of Kubernetes information.

i. Obtain information about pods.

~/.kube » kubectl get pods				
NAME	READY	STATUS	RESTARTS	AGE
nginx-log-demo-0-bxl79	1/1	Running	0	48d
nginx-log-demo-1-qmrqk	1/1	Running	0	48d
nginx-log-demo-2-7khv9	1/1	Running	0	48d
nginx-log-demo-3-j24xc	1/1	Running	0	48d

ii. Obtain information about namespaces.



2. Create a Logtail configuration.

Example:

```
{
    "inputs": [
        {
            "type": "service_docker_stdout",
            "detail": {
                 "Stdout": true,
                 "Stderr": true,
                "K8sNamespaceRegex":"^(default)$",
                "K8sPodRegex":"^(nginx-log-demo-1)$",
                "K8sContainerRegex":"^(nginx-log-demo-0)$"
        }
    }
}
```

Example 4: Filter containers by using Kubernetes labels

Collect stdout and stderr from containers whose Kubernetes labels contain the job-name key and a specific value. The value starts with nginx-logdemo.

1. Obtain Kubernetes labels.



2. Create a Logtail configuration

Example:

Examples of Logtail configurations for multi-line logs

Java exception stack logs are multi-line logs. You can create a Logtail configuration to collect the Java exception stack logs based on the following descriptions:

Sample logs

2021-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : service start 2021-02-03 14:18:41.969 ERROR [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : java.lang.NullPointerException at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193) at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166) at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:199) at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96) ... 2021-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : service start done

Logtail configuration

Collect the Java exception stack logs of the containers whose container label is app=monitor. The Java exception stack logs start with a date that is in a fixed format. Logtail matches only the first 10 bytes of each line to improve match efficiency. After the logs are collected and sent to Log Service, Log Service uses regular expressions to parse the logs into fields such as **time**, **leve**, **module**, **thread**, and **message**.

• inputs is required and is used to configure the data collection settings for the Logtail configuration. You must configure inputs based on your data source.

NoteYou can specify only one type of data source in inputs.

 processors is optional and is used to configure the data processing settings for the Logtail configuration. You can specify one or more processing methods. For more information, see Customize Logtail plug-ins to process data.

{
"inputs": [
(
"detail": {
"BeginLineCheckLength": 10,
"BeginLineRegex": "\\d+-\\d+-\\d+.*",
"IncludeLabel": {
"app": "monitor"
}
1,
"type": "service_docker_stdout"
}
],
"processors": [
(
"type": "processor_regex",
"detail": {
"SourceKey": "content",
$\label{eq:start} $$ Regex": $$ (\\d+-\\d+ \\d+:\\d+:\\d+\\.\\d+)\\s+(\[([^]]+)]\\s+(\[([^]]+)]\\s+([\\s\\S]*)", $$ (([\s\\S]*)", $$ ((([\s\\S]*)", $$ (([\s\\S]*)", "(([\s\S]*)", "(([([\s\\S]*)", "(([([\s\\S]*)", "(([([([\s\S]*)", "(([([([([([([([([(([(([(([(([(([(([(([($
"Keys": [
"time",
"level",
"module",
"thread",
"message"
1,
"NoKeyError": true,
"NoMatchError": true,
"KeepSource": false
}
}
1
)

Parsed logs

For example, if the collected log is 2018-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : service start done , the log is parsed into the following fields:

__tag_:__hostname__:logtail-dfgef _container_name_:example.com-hangzhou.aliyuncs.xxxxxxxxxxxxxx _namespace_:default _pod_name_:monitor-6f54bd5d74-rtzc7 _pod_uid_:7f012b72-04c7-11e8-84aa-00163f00c369 _source_:stdout _time_:2018-02-02T14:18:41.979147844Z time:2018-02-02 02:18:41.968 level:INFO module:spring-cloud-monitor thread:nio-8080-exec-4 class:c.g.s.web.controller.DemoController message:service start done

Log fields

The following table describes the fields that are uploaded by default for each log in a Kubernetes cluster.

Log field	Description
time	The time at which the data is uploaded. Example: 2021-02-02T02:18:41.979147844z
source	The type of the data source. Valid values: stdout and stderr.
_image_name_	The name of the image.
_container_name_	The name of the container.
_pod_name_	The name of the pod.
namespace	The namespace of the pod.
_pod_uid_	The unique identifier of the pod.
_container_id_	The IP address of the pod.

4.3.1.5.5. Use CRDs to collect container logs in DaemonSet mode

After you install Logtail in a container in DaemonSet mode, you can use a custom resource definition (CRD) to create a Logtail configuration and use the Logtail configuration to collect container logs.

Prerequisites

The alibaba-log-controller component is installed. For more information, see Install the Logtail component.

Implementation

The following list describes the process in which logs are collected by using a CRD:

- 1. The kubect1 tool or other tools are used to apply an AliyunLogConfig CRD.
- 2. The alibaba-log-controller detects the update in CRD configurations.
- The alibaba-log-controller sends requests to Log Service to create a Logstore, create a Logstail configuration, and apply the Logtail configuration to a machine group based on the content of the CRD and the status of the Logtail configurations in Log Service.
- Logtail periodically sends a request to the server on which the Logtail configuration is created to obtain the new or updated Logtail configuration and perform hot reloading.
- 5. Logtail collects stdout and stderr logs or text logs from each container based on the obtained Logtail configuration.
- 6. Logtail sends the collected container logs to Log Service.

Limits

- Limits on text log collection
 - If Logtail detects the die event on a container that is stopped, Logtail stops collecting text logs from the container. If collection latency occurs, some text logs that are generated before the container is stopped may be lost.
- Logtail cannot access the symbolic link of a container. You must specify an actual path as the collection directory.
- If a volume is mounted on the data directory of a container, Logtail cannot collect data from the parent directory of the data directory. You must
 specify the complete path of the data directory as the collection directory.
- For example, if a volume is mounted on the /var/log/service directory and you set the collection directory to /var/log, Logtail cannot collect logs from the /var/log directory. You must specify /var/log/service as the collection directory.
- By default, Kubernetes mounts the root directory of the host on the /logtail_host directory of the Logtail container. If you want to collect text logs from the host, you must specify /logtail_host as the prefix of the log file path.

For example, if you want to collect logs from the <code>/home/logs/app_log/</code> directory of the host, you must specify <code>/logtail_host/home/logs/app_log/</code> as the log file path.

- For Docker containers, only overlay and overlay2 storage drivers are supported. If other storage drivers are used, you must mount a volume on the directory of logs. Then, a temporary directory is generated.
- Limits on stdout and stderr log collection

The logging driver collects stdout and stderr logs only in the JSON format from containers that use the Docker engine.

General limits

Logtail collects data from containers that use the Docker engine or containerd engine.

- Docker: Logtail accesses the Docker engine in the /run/docker.sock directory. Make sure that the directory exists and Logtail has the permissions to
 access the directory.
- containerd: Logtail accesses the containerd engine in the /run/containerd/containerd.sock directory. Make sure that the directory exists and Logtail
 has the permissions to access the directory.

Create a Logtail configuration

To create a Logtail configuration, you need to only create an AliyunLogConfig CRD. After you create a Logtail configuration, the system automatically applies the Logtail configuration. If you want to delete the Logtail configuration, you need to only delete the CRD.

1. Log on to your Kubernetes cluster.

2. Run the following command to create a YAML file.

In this example, the file name is cube.yaml. Replace the file name with an actual file name.

vim cube.yaml

- 3. Enter the following script in the YAML file and configure the parameters based on your business scenario.
 - () Important
 - The value of the **configName** parameter must be unique in the Log Service project that you use.
 - If multiple CRDs are associated with the same Logtail configuration, the Logtail configuration is affected when you delete or modify one of the CRDs. After the deletion or modification, the status of the other CRDs that are associated with the Logtail configuration becomes inconsistent with the status of the Logtail configuration in Log Service.

apiVersion: log.alibabacloud.com/v1alpha1	# The default value is used. You do not need to modify this parameter.
kind: AliyunLogConfig	\sharp The default value is used. You do not need to modify this parameter.
metadata:	
name: simple-stdout-example	\sharp The name of the resource. The name must be unique in the current Kubernetes cluster.
spec:	
project: k8s-my-project	# Optional. The name of the project. The default value is the name of the project that you use
to install the Logtail component.	
logstore: k8s-stdout	\sharp The name of the Logstore. If the Logstore that you specify does not exist, Log Service autom
atically creates a Logstore.	
shardCount: 2	# Optional. The number of shards. Valid values: 1 to 10. Default value: 2.
lifeCycle: 90	\sharp Optional. The data retention period of the Logstore. The value of this parameter takes effect
only when you create a Logstore. Valid values:	1 to 3650. Unit: days. Default value: 90. The value 3650 specifies that log data is
permanently stored in the Logstore.	
logtailConfig:	# The Logtail configuration.
inputType: plugin	# The type of the data source. Valid values: file and plugin. file specifies text logs. plugin
specifies stdout and stderr logs.	
configName: simple-stdout-example	\sharp The name of the Logtail configuration. The name must be the same as the resource name that
is specified in metadata.name.	
inputDetail:	# The detailed settings of the Logtail configuration. For more information, see the following
and investigation and and	

Required Parameter Туре Description The name of the project. The default value is the name of the project that you use to install the Logtail component. string project No The name of the Logstore. logstore string Yes If the Logstore that you specify does not exist, Log Service automatically creates a Logstore shardCount int No The number of shards. Valid values: 1 to 10. Default value: 2. The data retention period of the Logstore. Valid values: 1 to 3650. Unit: days. Default value: 90. The value 3650 specifies that log data is permanently stored in the Logstore. ! Important lifeCycle int No The value of this parameter takes effect only when you create a Logstore. If you change the value of the lifeCycle parameter for an existing Logstore that is specified by the **logstore** parameter, the new value does not take effect. The machine group to which the Logtail configuration is applied. The default value is the machine group named k8s-group-\${your_k8s_cluster_id} . This machine group is automatically created machineGroups array No by Log Service when you install the Logtail component. The detailed settings of the Logtail configuration. In most cases, you need to configure only the **inputType**, **configName**, and **inputDetail** parameters. logtailConfig object Yes

4. Run the following command to apply the Logtail configuration.

In this example, the file name is cube.yaml. Replace the file name with an actual file name.

kubectl apply -f cube.yaml

After the Logtail configuration is applied, Logtail collects stdout and stderr logs or text logs from each container, and then sends the collected logs to Log Service.

View Logtail configurations

You can view Logtail configurations in the Log Service console or by using CRDs. For more information about how to view Logtail configurations in the Log Service console, see Manage a Logtail configuration.

() Important

If you modify the settings of a Logtail configuration in the Log Service console and view the Logtail configuration by using a CRD, the modification is not displayed in the output of the CRD. If you modify the settings of a Logtail configuration by using a CRD and view the Logtail configuration in the Log Service console, the modification is displayed in the Log Service console.

View all Logtail configurations in the current Kubernetes cluster

You can run the kubectl get aliyunlogconfigs command to view all Logtail configurations. The following figure shows the output.

shell@Alicloud:~\$ kubectl get aliyunlogconfigs NAME AGE docker-stdout 27m shell@Alicloud:~\$

View the details and status of a Logtail configuration

You can run the kubectl get aligunlogconfigs config_name -o yaml command to view the details and status of a Logtail configuration. The config_name parameter in the command specifies the name of the Logtail configuration that you want to view. Replace the configuration name with an actual configuration name. The following figure shows the output.

The **status** and **statusCode** parameters in the output indicate the status of the Logtail configuration.

- If the value of the **statusCode** parameter is 200, the Logtail configuration is applied.
- If the value of the **statusCode** parameter is not 200, the Logtail configuration fails to be applied.

shell@Alicloud:~\$ kubectl get aliyunlogconfigs docker-stdout -o yaml
apiVersion: log.alibabacloud.com/vlalphal
kind: AliyunLogConfig
metadata:
annotations:
kubectl.kubernetes.io/last-applied-configuration:
{"apiVersion":"log.alibabacloud.com/vlalphal","kind":"AliyunLogConfig","metadata":{"annotations":{},"name":"docker-stdout","namespace":"default"},"spec":{"logstore":"
cube-stdout","logtailConfig":{"configName":"docker-stdout","inputDetail":{"plugin":{"inputs":[{"detail":{"Stderr":true,"Stdout":true},"type":"service_docker_stdout"}]}},"in
putType":"plugin"}}}
creationTimestamp: "2021-10-29T08:40:332"
generation: 2
name: docker-stdout
namespace: default
resourceVersion: "1350968"
uid: 35f9516b 59fabdc
spec:
extenions: ""
lifeCycle: null
logstore: cube-stdout
logtailConfig:
configName: docker-stdout
inputDetail:
plugin:
inputs:
- detail:
Stderr: true
Stdout: true
type: service docker stdout
inputType: plugin
machineGroups: null
productCode: ""
productLanguage: ""
project: ""
shardCount: null
status:
status: OK
statusCode: 200

Examples of Logtail configurations that are used to collect stdout and stderr logs

If you want to collect container stdout and stderr logs, you must set the **inputType** parameter to plugin and add detailed settings to the plugin field of the inputDetail parameter. For more information about the parameters and the descriptions of the parameters, see Use the Log Service console to collect container stdout and stderr in DaemonSet mode.

Example 1: Collect container stdout and stderr logs in simple mode

Collect stdout and stderr logs from all containers except the containers whose environment variable configurations include COLLECT_STDOUT_FLAG=false configuration example:

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # The name of the resource. The name must be unique in the current Kubernetes cluster.
 name: simple-stdout-example
spec:
  # The name of the Logstore. If the Logstore that you specify does not exist, Log Service automatically creates a Logstore.
  logstore: k8s-stdout
  # The Logtail configuration.
  logtailConfig:
    # The type of the data source. If you want to collect stdout and stderr logs, you must set the value to plugin.
    inputType: plugin
    # The name of the Logtail configuration. The name must be the same as the resource name that is specified in metadata.name.
    configName: simple-stdout-example
    inputDetail:
      plugin:
        inputs:
            # input type
            type: service_docker_stdout
            detail:
              \ensuremath{\#} The settings that allow Logtail to collect both stdout and stderr logs.
             Stdout: true
              Stderr: true
              # The environment variable denylist. In this example, stdout and stderr logs are collected from all containers except the
containers whose environment variable configurations include COLLECT_STDOUT_FLAG=false.
             ExcludeEnv:
               COLLECT STDOUT FLAG: "false"
```

Example 2: Collect container stdout and stderr logs in simple mode and process the logs by using regular expressions

To view the environment variables of a container, you can log on to the host on which the container resides.

CRD configuration

apiVersion: log.alibabacloud.com/v1alpha1 kind: AliyunLogConfig metadata: # The name of the resource. The name must be unique in the current Kubernetes cluster. name: regex-stdout-example spec: # The name of the Logstore. If the Logstore that you specify does not exist, Log Service automatically creates a Logstore. logstore: k8s-stdout-regex # The Logtail configuration. logtailConfig: # The type of the data source. If you want to collect stdout logs, you must set the value to plugin. inputType: plugin # The name of the Loqtail configuration. The name must be the same as the resource name that is specified in metadata.name. configName: regex-stdout-example inputDetail: plugin: inputs: # input type type: service_docker_stdout detail: # The settings that allow Logtail to collect only stdout logs. Stdout: true Stderr: false # The environment variable allowlist. In this example, stdout logs are collected only from containers whose environment variabl e configurations include a key of GF_INSTALL_PLUGINS. IncludeEnv: GF_INSTALL_PLUGINS: '' processors: # The settings that allow Logtail to parse the collected stdout logs by using a regular expression. type: processor_regex detail: # The name of the source field. By default, the collected stdout logs are stored in the content field. SourceKey: content # The regular expression that is used to extract log content. $\texttt{Regex: 't=(\d+-\d+-\d++\d++\d+) lvl=(\w+) msg="([^{"}]+)" logger=(\w+) userId=(\w+) uname=(\S^*) method=(\w+) patrix and a state of the state o$ $h=(\S+)$ status=(\d+) remote addr=(\S+) time ms=(\d+) size=(\d+) referer=(\S*).*' # The keys that you want to extract from logs Keys: ['time', 'level', 'message', 'logger', 'userId', 'orgId', 'uname', 'method', 'path', 'status', 'remote_addr', 'time_ms', 'size', 'referer'] # The settings that allow Logtail to retain the source field. KeepSource: tru # The settings that allow Logtail to report an error when the specified source field does not exist. NoKeyError: tru # The settings that allow Logtail to report an error when the specified regular expression does not match the value of the spec ified source field. NoMatchError: true

Raw log

t=2018-03-09T07:14:03+0000 lvl=info msg="Request Completed" logger=context userId=0 orgId=0 uname= method=GET path=/ status=302 remote addr=172.16.64.154 time ms=0 size=29 referer=

Parsed log

05-11 20:10:16	_source_: 1
	tag:_hostname_: iZbp1p9rZ
	tag:_path_: /log/error.log
	topic:
	file : SessionTrackerImpl.java
	level : INFO
	line: 148
	message : Expiring sessions
	java.sql.SQLException: Incorrect string value: '\xF0\x9F\x8E\x8F",' for column 'data' at row 1
	at org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFallbackSQLExceptionTranslator.java:84)
	at org.springframework.jdbc.support.AbstractFallbackSQLException
	method : SessionTracker
	time: 2018-05-11T20:10:16,000

Examples of Logtail configurations that are used to collect text logs

If you want to collect container text logs, you must set the inputType parameter to file and add detailed settings to the inputDetail parameter. For more information about the parameters and the descriptions of the parameters, see Use the Log Service console to collect container text logs in DaemonSet mode.

Example 1: Collect container text logs in simple mode

Collect container text logs whose environment variable configurations include a key of ALIYUN_LOGTAIL_USER_DEFINED_ID. The log file path is /data/logs/app_1/simple.LOG.

apiVersion: log.alibabacloud.com/v1alpha1 kind: AliyunLogConfig metadata: # The name of the resource. The name must be unique in the current Kubernetes cluster. name: simple-file-example spec: # The name of the Logstore. If the Logstore that you specify does not exist, Log Service automatically creates a Logstore. logstore: k8s-file # The Logtail configuration logtailConfig: # The type of the data source. If you want to collect text logs, you must set the value to file. inputType: file # The name of the Logtail configuration. The name must be the same as the resource name that is specified by the metadata.name parameter. configName: simple-file-example inputDetail: # The settings that allow Logtail to collect text logs in simple mode. logType: common_reg_log # The log file path. logPath: /data/logs/app_1 # The log file name. You can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file name. Ex ample: log_*.log. filePattern: simple.LOG # If you want to collect container text logs, you must set the dockerFile parameter to true. dockerFile: true # The environment variable allowlist. In this example, text logs are collected only from containers whose environment variable configurations include a key of ALIYUN_LOGTAIL_USER_DEFINED_ID. dockerIncludeEnv:

ALIYUN LOGTAIL USER DEFINED ID: ""

Example 2: Collect container text logs in full regex mode

A Java program generates a multi-line log that contains error stack information. You can collect the log in full regex mode and specify a regular expression that is used to match the start part in the first line of the log in the Logtail configuration.

· Sample log

```
[2018-05-11T20:10:16,000] [INFO] [SessionTracker] [SessionTrackerImpl.java:148] Expiring sessions
   java.sql.SQLException: Incorrect string value: '\xF0\x9F\x8E\x8F",...' for column 'data' at row 1
   at org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFallbackSQLExceptionTranslator.java:84)
   at org.springframework.jdbc.support.AbstractFallbackSQLException

    CRD configuration

   apiVersion: log.alibabacloud.com/v1alpha1
   kind: AliyunLogConfig
   metadata:
     # The name of the resource. The name must be unique in the current Kubernetes cluster.
     name: regex-file-example
   spec:
     # The name of the Logstore. If the Logstore that you specify does not exist, Log Service automatically creates a Logstore.
     logstore: k8s-file
     logtailConfig:
       # The type of the data source. If you want to collect text logs, you must set the value to file.
       inputType: file
       # The name of the Logtail configuration. The name must be the same as the resource name that is specified in metadata.name.
       configName: regex-file-example
       inputDetail:
         # The settings that allow Logtail to collect text logs in full regex mode.
         logType: common reg log
         # The log file path.
         logPath: /app/logs
         # The log file name. You can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file name.
   Example: log *.log.
         filePattern: error.LOG
         # The regular expression that is used to match the start part in the first line of the log.
         logBeginRegex: '\[\d+-\d+-\w+:\d+:\d+,\d+]\s\[\w+]\s.*
         # The regular expression that is used to extract log content.
         regex: '\[([^]]+)]\s\[(\w+)]\s\[(\w+)]\s\[([^:]+):(\d+)]\s(.*)'
         # The keys that you want to extract from logs.
         key : ["time", "level", "method", "file", "line", "message"]
         # The format of the time values that are extracted from logs. By default, time values are extracted from the time field of logs that a
   re collected in full regex mode. If you do not want to extract time values, you can leave this parameter empty. If you configure the
   timeFormat parameter, you must also configure the adjustTimezone and logTimezone parameters
         timeFormat: '%Y-%m-%dT%H:%M:%S'
         # By default, Logtail uses UTC. You must configure the following parameter before you can forcefully change the time zone:
         adjustTimezone: true
         # The time zone offset. The time zone of logs is UTC+8. You can change the value of this parameter to change the time zone.
         logTimezone: "GMT+08:00"
         # The settings that allow Logtail to upload raw logs if the logs fail to be parsed.
         discardUnmatch: false
         # If you want to collect container text logs, you must set the dockerFile parameter to true.
         dockerFile: true
         # The environment variable allowlist. In this example, text logs are collected only from containers whose environment variable
   configurations include a key of ALIYUN_LOGTAIL_USER_DEFINED_ID.
         dockerIncludeEnv:
           ALIYUN_LOGTAIL_USER_DEFINED_ID: ""
```

Collected log

05-11 20:10:16	_source_: 10	
	tag : hostname_: iZbp14jp9rZ	
_tag :_path_: /log/error.log		
	topic :	
	file: SessionTrackerImpl.java	
	level : INFO	
	line: 148	
	message : Expiring sessions	
	java.sql.SQLException: Incorrect string value: '\xF0\x9F\x8E\x8F",' for column 'data' at row 1	
	at org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFallbackSQLExceptionTranslator.java:84)	
	at org.springframework.jdbc.support.AbstractFallbackSQLException	
	method : SessionTracker	
	time: 2018-05-11T20:10:16,000	

Example 3: Collect container text logs in delimiter mode

If the container text logs that you want to collect contain delimiters, you can collect the container text logs in delimiter mode. Logs that are in the delimiter-separated values (DSV) format use line feeds as boundaries. Each log is placed in a separate line. Each log is parsed into multiple fields by using delimiters.

apiVersion: log.alibabacloud.com/v1alpha1 kind: AliyunLogConfig metadata: # The name of the resource. The name must be unique in the current Kubernetes cluster. name: delimiter-file-example spec: # The name of the Logstore. If the Logstore that you specify does not exist, Log Service automatically creates a Logstore. logstore: k8s-file logtailConfig: # The type of the data source. If you want to collect text logs, you must set the value to file. inputType: file configName: delimiter-file-example # The name of the Logtail configuration. The name must be the same as the resource name that is specified in metadata.name. inputDetail: # The settings that allow Logtail to collect text logs in delimiter mode. logType: delimiter log # The log file path. logPath: /usr/local/ilogtail # The log file name. You can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file name. Ex ample: log *.log. filePattern: delimiter log.LOG # The delimiter. separator: '|&|' # The keys that you want to extract from logs key : ['time', 'level', 'method', 'file', 'line', 'message'] # The name of the field from which time values are extracted. timeKey: 'time' # The format of the time values that are extracted from logs. By default, time values are extracted from the time field of logs that are collected in delimiter mode. If you do not want to extract time values, you can leave this parameter empty. If you configure the timeFormat pa rameter, you must also configure the adjustTimezone and logTimezone parameters. timeFormat: '%Y-%m-%dT%H:%M:%S' # By default, Logtail uses UTC. You must configure the following parameter before you can forcefully change the time zone: adjustTimezone: true # The time zone offset. The time zone of logs is UTC+8. You can change the value of this parameter to change the time zone. logTimezone: "GMT+08:00" # The settings that allow Logtail to upload raw logs if the logs fail to be parsed. discardUnmatch: false # If you want to collect container text logs, you must set the dockerFile parameter to true. dockerFile: true # The environment variable allowlist. In this example, text logs are collected only from containers whose environment variable configurations include a key of ALIYUN_LOGTAIL_USER_DEFINED_ID. dockerIncludeEnv: ALIYUN_LOGTAIL_USER_DEFINED_ID: ''

Example 4: Collect container text logs in JSON mode

If the container text logs that you want to collect are JSON logs of the object type, you can collect the container text logs in JSON mode.

Raw log

{"url": "POST /PutData?

CRD configuration

apiVersion: log.alibabacloud.com/v1alpha1 kind: AliyunLogConfig metadata: # The name of the resource. The name must be unique in the current Kubernetes cluster. name: json-file-example spec: # The name of the Logstore. If the Logstore that you specify does not exist, Log Service automatically creates a Logstore. logstore: k8s-file logtailConfig: # The type of the data source. If you want to collect text logs, you must set the value to file. inputType: file # The name of the Logtail configuration. The name must be the same as the resource name that is specified in metadata.name. configName: json-file-example inputDetail: # The settings that allow Logtail to collect text logs in JSON mode. logType: json_log # The log file path. logPath: /usr/local/ilogtail # The log file name. You can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file name. Example: log_*.log. filePattern: json_log.LOG # The name of the field from which time values are extracted. If no requirements are specified, set the value to timeFormat: ''. timeKey: 'time' # The format of the time values that are extracted from logs. If no requirements are specified, set the value to timeFormat: ''. timeFormat: '%Y-%m-%dT%H:%M:%S' # If you want to collect container text logs, you must set the dockerFile parameter to true. dockerFile: true # The environment variable allowlist. In this example, text logs are collected only from containers whose environment variable configurations include a key of ALIYUN_LOGTAIL_USER_DEFINED_ID. dockerIncludeEnv:

ALIYUN_LOGTAIL_USER_DEFINED_ID: ""

4.3.1.5.6. Use CRDs to collect container text logs in Sidecar mode

This topic describes how to install Sidecar. This topic also describes how to use a custom resource definition (CRD) to create a Logtail configuration that is used to collect container text logs in Sidecar mode.

Prerequisites

The alibaba-log-controller component is installed. For more information, see Install the Logtail component.

Background information

In Sidecar mode, the Logtail container shares a log directory with an application container. The application container writes logs to the shared directory. Logtail monitors changes to the log files in the shared directory and collects logs. For more information, see Sidecar container with a logging agent and How Pods manage multiple containers.

Step 1: Install Sidecar

1. Log on to your Kubernetes cluster.

2. Create a YAML file.

In this command, the file name is sidecar.yaml. Replace the file name with an actual file name.

vim sidecar.yaml

3. Enter the following script in the YAML file and configure the parameters based on your business scenario.

() Important

Make sure that the time zone you specify for the *TZ* field in the *env* parameter is valid. If the time zones in raw logs and processed logs in a Log Service project are inconsistent, the time that is recorded for the collected logs may be a point in time in the past or in the future. For example, if the Log Service project resides in greater China, you can set the time zone to Asia/Shanghai.

apiVersion: batch/v1 kind: Job metadata: name: nginx-log-sidecar-demo namespace: default spec: template: metadata: name: nginx-log-sidecar-demo spec: restartPolicy: Never containers: - name: nginx-log-demo image: registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest command: ["/bin/mock_log"] args: ["--log-type=mginx", "--stdout=false", "--stderr=true", "--path=/var/log/nginx/access.log", "--total-count=1000000000", "--logs -per-sec=100"] volumeMounts: - name: nginx-log mountPath: /var/log/nginx ##### logtail sidecar container - name: logtail # more info: https://cr.console.aliyun.com/repository/cn-hangzhou/log-service/logtail/detail # this images is released for every region image: registry.cn-hangzhou.aliyuncs.com/log-service/logtail:latest $\ensuremath{\#}$ when recevie sigterm, logtail will delay 10 seconds and then stop command: - sh - -c - /usr/local/ilogtail/run_logtail.sh 10 livenessProbe: exec: command: - /etc/init.d/ilogtaild - status initialDelaySeconds: 30 periodSeconds: 30 resources: limits: memory: 512Mi requests: cpu: 10m memory: 30Mi env: ##### base config # user id - name: "ALIYUN LOGTAIL USER ID" value: "\${your_aliyun_user_id}" # user defined id - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID" value: "\${your_machine_group_user_defined_id}" # config file path in logtail's container - name: "ALIYUN LOGTAIL CONFIG" value: "/etc/ilogtail/conf/\${your region config}/ilogtail config.json" ##### env tags config - name: "ALIYUN_LOG_ENV_TAGS" value: "_pod_name_|_pod_ip_|_namespace_|_node_name_|_node_ip_" - name: "_pod_name_" valueFrom: fieldRef: fieldPath: metadata.name - name: "_pod_ip_" valueFrom: fieldRef: fieldPath: status.podIP - name: "_namespace_" valueFrom: fieldRef: fieldPath: metadata.namespace - name: "_node_name_" valueFrom: fieldRef: fieldPath: spec.nodeName - name: "_node_ip_" valueFrom: fieldRef: fieldPath: status.hostIP volumeMounts: - name: nginx-log mountPath: /var/log/nginx ##### share this volume volumes: - name: nginx-log emptyDir: {}

i. Configure the basic variables in the configuration script. The following table describes the variables.

base config

- # user id
- name: "ALIYUN_LOGTAIL_USER_ID" value: "\${your_aliyun_user_id}"
- # user defined id
- name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
- value: "\${your_machine_group_user_defined_id}"
 # config file path in logtail's container
- name: "ALIYUN_LOGTAIL_CONFIG"
 - value: "/etc/ilogtail/conf/\${your_region_config}/ilogtail_config.json"

Variable	Description
\${your_aliyun_user_id}	The ID of your Apsara Stack tenant account. For more information, seeConfigure a user identifier.
\${your_machine_group_user_defined_id}	The custom identifier of your machine group. The identifier must be unique in the region where your project resides. Example: nginx-log-sidecar. For more information, see Create a custom identifier-based machine group.
\${your_region_config}	The ID of the region where your project resides and the type of the network that your project uses. For more information about regions, see Manage a Logtail configuration.

ii. Specify the mount path in the configuration script.

⑦ Note

We recommend that you mount containers on a volume of the emptyDir type.

volumeMounts: - name: nginx-log mountPath: /var/log/nginx ##### share this volume volumes: - name: nginx-log

emptyDir: {}

Parameter	Description		
	The name of the volume. You can specify a name based on your business requirements.		
name	() Important The value of the name parameter in the volumeMounts node and the value of thename parameter in the volumes node must be the same. This ensures that the Logtail container and the application container are mounted on the same volume.		
mountPath	The mount path. You can enter the path of files in which container text logs are recorded.		

iii. Specify a waiting period for the Logtail container in the configuration script.

In most cases, the waiting period is 10 seconds. This value specifies that the Logtail container exits 10 seconds after the container receives a stop command. This setting helps prevent incomplete data collection.

C	ommand:
-	sh
-	-c

- /usr/local/ilogtail/run_logtail.sh 10
- 4. Run the following command to apply the configurations in the sidecar.yaml file.

In this command, the file name is sidecar.yaml. Replace the file name with an actual file name.

kubectl apply -f sidecar.yaml

Step 2: Create a Logtail configuration

To create a Logtail configuration, you only need to create an AliyunLogConfig CRD. After you create a Logtail configuration, the system automatically applies the Logtail configuration. If you want to delete the Logtail configuration, you only need to delete the CRD.

1. Log on to your Kubernetes cluster.

2. Run the following command to create a YAML file.

In this command, the file name is cube.yaml. Replace the file name with an actual file name.

vim cube.yaml

3. Enter the following script in the YAML file and configure the parameters based on your business scenario.

() Important

• The value of the **configName** parameter must be unique in the Log Service project that you use.

 If multiple CRDs are associated with the same Logtail configuration, the Logtail configuration is affected when you delete or modify one of the CRDs. After the deletion or modification, the status of the other associated CRDs becomes inconsistent with the status of the Logtail configuration in Log Service.

• In Sidecar mode, only text logs can be collected. You must set the **dockerFile** parameter to false.

apiVersion: log.alibabacloud.com/v1alpha1 kind: AliyunLogConfig metadata:		<pre># The default value is used. You do not need to modify this parameter. # The default value is used. You do not need to modify this parameter.</pre>			
name: simple-stdout-example spec:		\sharp The name of the resource. The name must be unique in the current Kubernetes cluster.			
project: k8s-my-project to install the Logtail component.		\sharp Optional. The name of the project. The default value is the name of the project that you use			
logstore: k8s-stdout atically creates a Logsto	# Th	$\ensuremath{\sharp}$ The name of the Logstore. If the Logstore that you specify does not exist, Log Service autom			
<pre>machineGroups: \${your_machine_group_user</pre>	# Th defined_id} parameter tha	e name of the machine grou t you configured when you	p. The name must be the same as the value of the installed Sidecar. This machine group is used to associate		
Sidecar with the CRD. - nginx-log-sidecar	# OT	tional The number of char	de Welid welves, 1 to 10 Default welves 2		
lifeCycle: 90 ault value: 90. The value	# 0g # 0g 2 3650 specifies that log o	tional. The data retention ata is permanently stored	period of the Logstore. Valid values: 1 to 3650. Unit: days. Def in the Logstore.		
logtailConfig: inputType: file	# Th # Th	e Logtail configuration. e type of the data source.	In Sidecar mode, you can use CRDs to collect only text logs. The		
refore, you must set the configName: simple-st	value to file. dout-example # Th	e name of the Logtail conf	iguration. The name must be the same as the resource name that		
is specified in metadata.name. inputDetail:		# The detailed settings of the Logtail configuration. For more information, see the following			
Parameter	Туре	Required	Description		
project	string	No	The name of the project. The default value is the name of the project that you use to install the Logtail component.		
la vatava	string	Vec	The name of the Logstore.		
logstore	sung	Tes	If the Logstore that you specify does not exist, Log Service automatically creates a Logstore.		
shardCount	int	No	The number of shards. Valid values: 1 to 10. Default value: 2.		
lifeCycle	int	No	The data retention period of the Logstore. Valid values: 1 to 3650. Unit: days. Default value: 90. The value 3650 specifies that log data is permanently stored in the Logstore.		

machineGroups	array	Yes	① Important You must specify a custom identifier for the machine group in the following format:
			machineGroups: - nginx-log-sidecar
logtailConfig	object	Yes	The detailed settings of the Logtail configuration. In most cases, you need to configure only the inputType , configName , and inputDetail parameters.

4. Run the following command to apply the Logtail configuration.

In this command, the file name is cube.yaml. Replace the file name with an actual file name.

kubectl apply -f cube.yaml

After you create the Logtail configuration, you can view the Logtail configuration in the Log Service console or by using a CRD. For more information, see Manage a Logtail configuration.

Configuration example for a single directory

This section provides an example on how to use a CRD to collect text logs from the nginx-log-demo container in Sidecar mode. The container belongs to a self-managed Kubernetes cluster in a data center. The text logs include NGINX access logs and NGINX error logs and are stored in a single directory. The following list describes the basic information:

• The Log Service project for log collection resides in the China (Hangzhou) region. Logs are collected over the Internet.

Cloud Defined Storage

- The name of the volume to be mounted is **nginx-log** and the volume is of the **emptyDir** type. The **nginx-log** volume is mounted on the /var/log/nginx directory of the nginx-log-demo and Logtail containers.
- The path to NGINX access logs is /var/log/nginx/access.log. The name of the Logstore that is used to store the NGINX access logs is nginx-access.
- The path to NGINX error logs is /var/log/nginx/error.log. The name of the Logstore that is used to store the NGINX error logs is nginx-error.

```
• Sidecar configuration example
   apiVersion: batch/v1
   kind: Job
   metadata:
    name: nginx-log-sidecar-demo
    namespace: default
   spec:
     template:
       metadata:
        name: nginx-log-sidecar-demo
       spec:
         restartPolicy: Never
         containers:
         - name: nginx-log-demo
           image: registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest
           command: ["/bin/mock_log"]
           args: ["--log-type=nginx", "--stdout=false", "--stderr=true", "--path=/var/log/nginx/access.log", "--total-count=1000000000", "--logs
   -per-sec=100"]
          volumeMounts:
           - name: nginx-log
             mountPath: /var/log/nginx
         ##### logtail sidecar container
         - name: logtail
          # more info: https://cr.console.alivun.com/repository/cn-hangzhou/log-service/logtail/detail
           # this images is released for every region
           image: registry.cn-hangzhou.aliyuncs.com/log-service/logtail:latest
           \ensuremath{\#} when recevie sigterm, logtail will delay 10 seconds and then stop
           command:
           - sh
           - -c
           - /usr/local/ilogtail/run_logtail.sh 10
           livenessProbe:
             exec:
              command:
               - /etc/init.d/ilogtaild
               - status
             initialDelaySeconds: 30
             periodSeconds: 30
           env:
            ##### base config
             # user id
             - name: "ALIYUN_LOGTAIL_USER_ID"
               value: "1023****3423"
             # user defined id
             - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
               value: "nginx-log-sidecar"
             # config file path in logtail's container
             - name: "ALIYUN_LOGTAIL_CONFIG"
               value: "/etc/ilogtail/conf/cn-hangzhou-internet/ilogtail_config.json"
             ##### env tags config
              - name: "ALIYUN_LOG_ENV_TAGS"
              value: "_pod_name_|_pod_ip_|_namespace_|_node_name_|_node_ip_"
             - name: "_pod_name_"
              valueFrom:
                fieldRef:
                  fieldPath: metadata.name
             - name: "_pod_ip_"
               valueFrom:
                 fieldRef:
                  fieldPath: status.podIP
             - name: " namespace '
              valueFrom:
                 fieldRef:
                  fieldPath: metadata.namespace
             - name: "_node_name_"
               valueFrom:
                fieldRef:
                   fieldPath: spec.nodeName
             - name: "_node_ip_"
               valueFrom:
                 fieldRef:
                  fieldPath: status.hostIP
           volumeMounts:
           - name: nginx-log
             mountPath: /var/log/nginx
         ##### share this volume
         volumes:
          - name: nginx-log
           emptyDir: {}
```

CRD configuration example

Create a Logtail configuration to collect NGINX access logs and another Logtail configuration to collect NGINX error logs.

Collect NGINX access logs

() Important

In Sidecar mode, you must set the **dockerFile** parameter to false.

apiVersion: log.alibabacloud.com/v1alpha1

kind: AliyunLogConfig metadata:

The name of the resource. The name must be unique in your Kubernetes cluster.

name: nginx-log-access-example spec:

The name of the project. The default value is the name of the project that you use to install Logtail.

project: k8s-nginx-sidecar-dem

The name of the Logstore. If the Logstore that you specify does not exist, Log Service automatically creates a Logstore.

logstore: nginx-access

The name of the machine group. The name must be the same as the value of the \${your_machine_group_user_defined_id} parameter that you configured when you installed Sidecar.

machineGroups:

- nginx-log-sidecar # The Logtail configuration.

logtailConfig:

The type of the data source. In Sidecar mode, you can use CRDs to collect only text logs. Therefore, you must set the value to file. inputType: file

The name of the Logtail configuration. The name must be the same as the resource name that is specified in metadata.name.

- configName: nginx-log-access-example
- inputDetail:

The settings that allow Logtail to collect text logs in full regex mode.

logType: common_reg_log

The log file path.

logPath: /var/log/nginx

The log file name. You can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file name . Example: log_*.log.

filePattern: access.log

Set the dockerFile parameter to false. This setting is required in Sidecar mode.

dockerFile: false

The regular expression that is used to match the start part in the first line of the log. If you want to collect single-line logs, set the value to '.*'.

logBeginRegex: '.*'

The regular expression that is used to extract log content. Configure this parameter based on your business scenario.

The keys that you want to extract from logs.

key : ["time", "ip", "method", "url", "protocol", "latency", "payload", "status", "response-size",user-agent"]

• Collect NGINX error logs

() Important

In Sidecar mode, you must set the **dockerFile** parameter to false.

config for error log

apiVersion: log.alibabacloud.com/v1alpha1 kind: AliyunLogConfig metadata: # The name of the resource. The name must be unique in the current Kubernetes cluster. name: nginx-log-error-example spec: # The name of the project. The default value is the name of the project that you use to install Logtail. project: k8s-nginx-sidecar # The name of the Logstore. If the Logstore that you specify does not exist, Log Service automatically creates a Logstore. logstore: nginx-error # The name of the machine group. The name must be the same as the value of the \${your_machine_group_user_defined_id} parameter that you configured when you installed Sidecar. machineGroups: - nginx-log-sidecar # The Logtail configuration. logtailConfig: # The type of the data source. In Sidecar mode, you can use CRDs to collect only text logs. Therefore, you must set the value to file. inputType: file # The name of the Logtail configuration. The name must be the same as the resource name that is specified in metadata.name. configName: nginx-log-error-example inputDetail: # The settings that allow Logtail to collect text logs in full regex mode. logType: common_reg_log # The log file path. logPath: /var/log/nginx # The log file name. You can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file name . Example: log_*.log. filePattern: error.log # Set the dockerFile parameter to false. This setting is required in Sidecar mode. dockerFile: false

Configuration example for different directories

This section provides an example on how to use a CRD to collect text logs from the nginx-log-demo container in Sidecar mode. The container belongs to a self-managed Kubernetes cluster in a data center. The text logs include NGINX access logs and are stored in different directories. The following list describes the basic information:

- The Log Service project for log collection resides in the China (Hangzhou) region. Logs are collected over the Internet.
- The names of the volumes to be mounted are nginx-log and nginx-logs and the volumes are of the emptyDir type. The nginx-log volume is
 mounted on the /var/log/nginx directory of the nginx-log-demo and Logtail containers. The nginx-logs volume is mounted on the /var/log/nginxs
 directory of the nginx-log-demo and Logtail containers.
- One log file path is /var/log/nginx/access.log and the other log file path is /var/log/nginxs/access.log.
- The name of the Logstore that is used to store NGINX access logs is nginx-access.
- Sidecar configuration example

apiVersion: batch/v1 kind: Job metadata: name: nginx-log-sidecar-demo namespace: default spec: template: metadata: name: nginx-log-sidecar-demo spec: restartPolicy: Never containers: - name: nginx-log-demo image: registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest command: ["/bin/mock_log"] args: ["--log-type=nginx", "--stdout=false", "--stderr=true", "--path=/var/log/nginx/access.log", "--total-count=1000000000", "--logs -per-sec=100"] lifecycle: volumeMounts: - name: nginx-log mountPath: /var/log/nginx - name: nginx-logs mountPath: /var/log/nginxs ##### logtail sidecar container - name: logtail # more info: https://cr.console.aliyun.com/repository/cn-hangzhou/log-service/logtail/detail # this images is released for every region image: registry.cn-hangzhou.aliyuncs.com/log-service/logtail:latest $\ensuremath{\#}$ when recevie sigterm, logtail will delay 10 seconds and then stop lifecycle: command: - sh - -c - /usr/local/ilogtail/run logtail.sh 10 livenessProbe: exec: command: - /etc/init.d/ilogtaild - status initialDelaySeconds: 30 periodSeconds: 30 resources: limits: memory: 512Mi requests: cpu: 10m memory: 30Mi env: ##### base config # user id - name: "ALIYUN LOGTAIL USER ID" value: "1023****3423" # user defined id - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID" value: "nginx-log-sidecar" # config file path in logtail's container - name: "ALIYUN_LOGTAIL_CONFIG" value: "/etc/ilogtail/conf/cn-hangzhou-internet/ilogtail_config.json" ##### env tags config - name: "ALIYUN_LOG_ENV_TAGS" value: "_pod_name_l_pod_ip_|_namespace_|_node_name_|_node_ip_"
- name: "_pod_name_" valueFrom: fieldRef: fieldPath: metadata.name - name: "_pod_ip_" valueFrom: fieldRef: fieldPath: status.podIP - name: "_namespace_" valueFrom: fieldRef: fieldPath: metadata.namespace - name: "_node_name_" - luor

fieldRef: fieldPath: spec - name: "_node_ip_" valueFrom:	
fieldPath: spec - name: "_node_ip_" valueFrom:	
- name: "_node_ip_" valueFrom:	. nodeName
valueFrom:	
fieldRef:	
fieldPath: stat	us.hostIP
volumeMounts:	
- name: nginx-log	
mountPath: /var/log/n	iginx
- name: nginx-logs	
mountPath: /var/log/n	ıginxs
##### share this volume	
volumes:	
- name: nginx-log	
emptyDir: {}	
- name: nginx-logs	
emptyDir: {}	
configuration example	
te two Logtail configurations	to collect NGINX access logs from different directories.
llect NGINX access logs from	the /var/log/nginx/access.log directory
Important	
In Sidecar mode, you must se	at the declarEile parameter to false
in Sidecar mode, you must se	et the dockerrne parameter to faise.
aniVersion. log alibabacloud	com/wlalnhal
cind: AlivunLogConfig	
netadata:	
# The name of the resource.	. The name must be unique in the current Kubernetes cluster.
name: nginx-log-access-exam	mple
spec:	a · ·
# The name of the project.	The default value is the name of the project that you use to install Logtail.
project: k8s-nginx-sidecar-	-demo
# The name of the Logstore.	. If the Logstore that you specify does not exist, Log Service automatically creates a Logstore.
logstore: nginx-access	
# The name of the machine g	group. The name must be the same as the value of the \${your machine group user defined id} parameter that yo
configured when you installed	i Sidecar.
machineGroups:	
- nginx-log-sidecar	
# The Logtail configuration	1.
logtailConfig:	
	purce. In Sidecar mode, you can use CRDs to collect only text logs. Therefore, you must set the value to file
# The type of the data so	
<pre># The type of the data so inputType: file</pre>	
<pre># The type of the data sc inputType: file # The name of the Logtail</pre>	configuration. The name must be the same as the resource name that is specified in metadata.name.
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc</pre>	L configuration. The name must be the same as the resource name that is specified in metadata.name. ress-example
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail:</pre>	l configuration. The name must be the same as the resource name that is specified in metadata.name. :ess-example
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all</pre>	l configuration. The name must be the same as the resource name that is specified in metadata.name. :ess-example .ow Logtail to collect text logs in full regex mode.
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_reg_log</pre>	l configuration. The name must be the same as the resource name that is specified in metadata.name. ress-example Yow Logtail to collect text logs in full regex mode.
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_reg_log # The log file path.</pre>	l configuration. The name must be the same as the resource name that is specified in metadata.name. ress-example low Logtail to collect text logs in full regex mode.
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_reg_log # The log file path. logPath: /var/log/nginx</pre>	l configuration. The name must be the same as the resource name that is specified in metadata.name. :ess-example low Logtail to collect text logs in full regex mode. ; ;
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_reg_log # The log file path. logPath: /var/log/nginx # The log file name. Yo</pre>	l configuration. The name must be the same as the resource name that is specified in metadata.name. ress-example low Logtail to collect text logs in full regex mode. ; ; u can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file na
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_reg_log # The log file path. logPath: /var/log/nginx # The log file name. Yo Example: log_*.log.</pre>	l configuration. The name must be the same as the resource name that is specified in metadata.name. ress-example low Logtail to collect text logs in full regex mode. g c uu can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file na
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_reg_log # The log file path. logPath: /var/log/nginx # The log file name. Yo Example: log_*.log. filePattern: access.log</pre>	l configuration. The name must be the same as the resource name that is specified in metadata.name. cess-example low Logtail to collect text logs in full regex mode. y control to collect text logs in full regex mode. y control text lo
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_reg_log # The log file path. logPath: /var/log/nginx # The log file name. Yo Example: log_*.log. filePattern: access.log # Set the dockerFile pa</pre>	l configuration. The name must be the same as the resource name that is specified in metadata.name. ress-example low Logtail to collect text logs in full regex mode. c c cu can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file na f irameter to false. This setting is required in Sidecar mode.
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_reg_log # The log file path. logFath: /var/log/nginx # The log file name. Yo Example: log_*log. filePattern: access.log # Set the dockerFile pa dockerFile: false</pre>	l configuration. The name must be the same as the resource name that is specified in metadata.name. ress-example low Logtail to collect text logs in full regex mode. ; c pu can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file na ; urameter to false. This setting is required in Sidecar mode.
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_reg_log # The log file path. logPath: /var/log/nginx # The log file name. Yo Example: log_*.log. filePattern: access.log # Set the dockerFile pa dockerFile: false # The regular expression }</pre>	<pre>l configuration. The name must be the same as the resource name that is specified in metadata.name. cess-example low Logtail to collect text logs in full regex mode. j c uu can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file na j urameter to false. This setting is required in Sidecar mode. un that is used to match the start part in the first line of the log. If you want to collect single-line log</pre>
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_reg_log # The log file path. logPath: /var/log/nginx # The log file name. Yo Example: log_*.log. filePattern: access.log # Set the dockerFile pa dockerFile: false # The regular expressions the value to '.*'.</pre>	I configuration. The name must be the same as the resource name that is specified in metadata.name. ress-example tow Logtail to collect text logs in full regex mode. y c u can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file na y rameter to false. This setting is required in Sidecar mode. In that is used to match the start part in the first line of the log. If you want to collect single-line log
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_reg_log # The log file path. logPath: /var/log/nginx # The log file name. Yo Example: log_*.log. filePattern: access.log # Set the dockerFile pa dockerFile: false # The regular expression et the value to '.*'. logBeginRegex: '.*'</pre>	l configuration. The name must be the same as the resource name that is specified in metadata.name. cess-example low Logtail to collect text logs in full regex mode. y c su can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file no y trameter to false. This setting is required in Sidecar mode. on that is used to match the start part in the first line of the log. If you want to collect single-line log
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_reg_log # The log file path. logPath: /var/log/nginx # The log file name. Yo Example: log_*.log. filePattern: access.log # Set the dockerFile pa dockerFile: false # The regular expressio et the value to '.*'. logBeginRegex: '.*' # The regular expressio</pre>	I configuration. The name must be the same as the resource name that is specified in metadata.name. ress-example low Logtail to collect text logs in full regex mode. ; ; ; ; ; ; ; ; ; ; ; ; ;
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_req_log # The log file path. logPath: /var/log/nginx # The log file name. Yo Example: log_*.log. filePattern: access.log # Set the dockerFile pa dockerFile: false # The regular expressio et the value to '.*'. logBeginRegex: '.*' # The regular expressio regex: '(\S+)\s(\S+)\s\</pre>	<pre>l configuration. The name must be the same as the resource name that is specified in metadata.name. cess-example low Logtail to collect text logs in full regex mode. f c vu can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file na f irrameter to false. This setting is required in Sidecar mode. on that is used to match the start part in the first line of the log. If you want to collect single-line log in that is used to extract log content. (S+\s\S+\s"(\S+)\s(\S+)\s((\S+)\s((\S+)\s(\S+)\s(\S+)\s"([^"]+)"\s.*'</pre>
<pre># The type of the data sc inputType: file # The name of the Logtail configName: nginx-log-acc inputDetail: # The settings that all logType: common_req_log # The log file path. logPath: /var/log/nginx # The log file name. Yo Example: log_*.log. filePattern: access.log # Set the dockerFile pa dockerFile: false # The regular expressio et the value to '.*'. logBeginRegex: '.*' # The regular expressio regex: '(\S+)\s((S+)\s(# The keys that you wan</pre>	<pre>cl configuration. The name must be the same as the resource name that is specified in metadata.name. cess-example low Logtail to collect text logs in full regex mode. j c u can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file na j irrameter to false. This setting is required in Sidecar mode. on that is used to match the start part in the first line of the log. If you want to collect single-line log in that is used to extract log contentS+\s\S+\s"(\S+)\s(\S+)\s+([^"]+)"\s+(\S+)\s(\S+)\s(\d+)\s(\S+)\s"([^"]+)"\s.*' it to extract from logs.</pre>

· Collect NGINX access logs from the /var/log/nginxs/access.log directory () Important In Sidecar mode, you must set the **dockerFile** parameter to false. apiVersion: log.alibabacloud.com/v1alpha1 kind: AliyunLogConfig metadata: # The name of the resource. The name must be unique in the current Kubernetes cluster. name: nginxs-log-access-example spec: # The name of the project. The default value is the name of the project that you use to install Logtail. project: k8s-nginx-sideca: # The name of the Logstore. If the Logstore that you specify does not exist, Log Service automatically creates a Logstore. logstore: nginxs-access # The name of the machine group. The name must be the same as the value of the \${your machine group user defined id} parameter that you configured when you installed Sidecar. machineGroups: - nginx-log-sidecar # The Logtail configuration. logtailConfig: # The type of the data source. In Sidecar mode, you can use CRDs to collect only text logs. Therefore, you must set the value to file. inputType: file # The name of the Logtail configuration. The name must be the same as the resource name that is specified in metadata.name. configName: nginxs-log-access-example inputDetail: # The settings that allow Logtail to collect text logs in full regex mode. logType: common_reg_log # The log file path logPath: /var/log/nginxs # The log file name. You can use wildcard characters such as asterisks (*) and question marks (?) when you specify the log file name . Example: log *.log. filePattern: access.log # Set the dockerFile parameter to false. This setting is required in Sidecar mode dockerFile: false # The regular expression that is used to match the start part in the first line of the log. If you want to collect single-line logs, set the value to '.*' logBeginRegex: '.*' # The regular expression that is used to extract log content. # The keys that you want to extract from logs. key : ["time", "ip", "method", "url", "protocol", "latency", "payload", "status", "response-size", user-agent"] # config for error log

4.3.1.5.7. Use the Log Service console to collect container text logs in Sidecar

mode

This topic describes how to install Sidecar. This topic also describes how to use the Log Service console to create a Logtail configuration that is used to collect container text logs in Sidecar mode.

Prerequisites

The alibaba-log-controller component is installed. For more information, see Install the Logtail component.

Background information

In Sidecar mode, the Logtail container shares a log directory with an application container. The application container writes logs to the shared directory. Logtail monitors changes to log files in the shared directory and collects logs. For more information, see Sidecar container with a logging agent and How Pods manage multiple containers.

Step 1: Install Sidecar

- 1. Log on to your Kubernetes cluster.
- 2. Create a YAML file.

In this command, the file name is sidecar.yaml. Replace the file name with an actual file name.

vim sidecar.yaml

3. Enter the following script in the YAML file and configure the parameters based on your business scenario.

! Important

Make sure that the time zone you specify for the TZ field in the env parameter is valid. If the time zones in raw logs and processed logs in a Log Service project are inconsistent, the time that is recorded for the collected logs may be a point in time in the past or in the future. For example, if the Log Service project resides in greater China, you can set the time zone to Asia/Shanghai.

apiVersion: batch/v1 kind: Job metadata: name: nginx-log-sidecar-demo namespace: default spec: template: metadata: name: nginx-log-sidecar-demo spec: restartPolicy: Never containers: - name: nginx-log-demo image: registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest command: ["/bin/mock_log"] args: ["--log-type=mginx", "--stdout=false", "--stderr=true", "--path=/var/log/nginx/access.log", "--total-count=1000000000", "--logs -per-sec=100"] volumeMounts: - name: nginx-log mountPath: /var/log/nginx ###### logtail sidecar container - name: logtail # more info: https://cr.console.aliyun.com/repository/cn-hangzhou/log-service/logtail/detail # this images is released for every region image: registry.cn-hangzhou.aliyuncs.com/log-service/logtail:latest $\ensuremath{\#}$ when recevie sigterm, logtail will delay 10 seconds and then stop command: - sh - -c - /usr/local/ilogtail/run_logtail.sh 10 livenessProbe: exec: command: - /etc/init.d/ilogtaild - status initialDelaySeconds: 30 periodSeconds: 30 resources: limits: memory: 512Mi requests: cpu: 10m memory: 30Mi env: ##### base config # user id - name: "ALIYUN LOGTAIL USER ID" value: "\${your_aliyun_user_id}" # user defined id - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID" value: "\${your_machine_group_user_defined_id}" # config file path in logtail's container - name: "ALIYUN LOGTAIL CONFIG" value: "/etc/ilogtail/conf/\${your_region_config}/ilogtail_config.json" ##### env tags config - name: "ALIYUN_LOG_ENV_TAGS" value: "_pod_name_|_pod_ip_|_namespace_|_node_name_|_node_ip_" - name: "_pod_name_" valueFrom: fieldRef: fieldPath: metadata.name - name: "_pod_ip_" valueFrom: fieldRef: fieldPath: status.podIP - name: "_namespace_" valueFrom: fieldRef: fieldPath: metadata.namespace - name: "_node_name_" valueFrom: fieldRef: fieldPath: spec.nodeName - name: "_node_ip_" valueFrom: fieldRef: fieldPath: status.hostIP volumeMounts: - name: nginx-log mountPath: /var/log/nginx ##### share this volume volumes: - name: nginx-log emptyDir: {}

i. Configure the basic variables in the configuration script. The following table describes the variables.

base config

- # user id - name: "ALIYUN_LOGTAIL_USER_ID"
 - value: "\${your_aliyun_user_id}"
- # user defined id
- name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
- value: "\${your_machine_group_user_defined_id}"
 # config file path in logtail's container
- name: "ALIYUN_LOGTAIL_CONFIG"
- value: "/etc/ilogtail/conf/\${your_region_config}/ilogtail_config.json"

Variable	Description
\${your_aliyun_user_id}	The ID of your Apsara Stack tenant account. For more information, seeConfigure a user identifier.
\${your_machine_group_user_defined_id}	The custom identifier of your machine group. The identifier must be unique in the region where your project resides. Example: nginx-log-sidecar. For more information, see Create a custom identifier-based machine group.
\${your_region_config}	The ID of the region where your project resides and the type of the network that your project uses. For more information about regions, see Manage a Logtail configuration.

ii. Specify the mount path in the configuration script.

⑦ Note

We recommend that you mount containers on a volume of the emptyDir type.

volumeMounts: - name: nginx-log mountPath: /var/log/nginx ##### share this volume volumes: - name: nginx-log

emptyDir: {}

Parameter	Description		
	The name of the volume. You can specify a name based on your business requirements.		
name	() Important The value of the name parameter in the volumeMounts node and the value of thename parameter in the volumes node must be the same. This ensures that the Logtail container and the application container are mounted on the same volume.		
mountPath	The mount path. You can enter the path of files in which container text logs are recorded.		

iii. Specify a waiting period for the Logtail container in the configuration script.

In most cases, the waiting period is 10 seconds. This value specifies that the Logtail container exits 10 seconds after the container receives a stop command. This setting helps prevent incomplete data collection.

command:				
-	sh			
-	-c			

- /usr/local/ilogtail/run_logtail.sh 10
- 4. Run the following command to apply the configurations in the sidecar.yaml file.

In this command, the file name is sidecar.yaml. Replace the file name with an actual file name.

kubectl apply -f sidecar.yaml

Step 2: Create a machine group

- 1. Log on to the Log Service console
- 2. In the Projects section, click the project that you use to install Logtail components.
- 3. In the left-side navigation pane, choose **Resources > Machine Groups**.
- 4. In the Machine Groups list, choose $_{\mathbb{R}}$ > Create Machine Group.
- 5. In the Create Machine Group panel, configure the parameters and click OK. The following table describes the parameters.

		Parameter	Description
--	--	-----------	-------------

Name	The name of the machine group. Important After the machine group is created, you cannot change the name of the machine group. Proceed with caution.			
Identifier	The identifier of the machine group. Select Custom ID .			
Торіс	The topic of the machine group. The topic is used to differentiate the logs that are generated by different servers.			
Custom Identifier	The custom identifier of the machine group. The identifier must be the same as the value of the <i>fyour_machine_group_use</i> r_defined_id} parameter that you configured when you installed Sidecar. Example: nginx-log-sidecar.			

Step 3: Create a Logtail configuration

1. Log on to the Log Service console

2. In the Import Data section, click RegEx - Text Log.

In this example, a Logtail configuration is created to collect text logs in full regex mode. For information about how to collect text logs in other modes, see Collect text logs.

3. Select a project and a Logstore. Then, click **Next**.

Select the project that you use to install Logtail components and the Logstore that you create.

4. Click Use Existing Machine Groups.

5. Select a machine group from the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

Select the machine group that you created in Step 2: Create a machine group.

```
() Important
```

If you enable a machine group immediately after you create the machine group, the heartbeat status of the machine group may be **FAIL**. This issue occurs because the machine group is not connected to Log Service. To resolve this issue, you can click **Automatic Retry**. If the issue persists, troubleshoot the issue. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail client?.

6. Create a Logtail configuration and click Next.

You can collect logs in simple mode, NGINX mode, delimiter mode, JSON mode, or full regex mode. For more information, see Collect text logs.

() Important

In Sidecar mode, do not turn on Docker File.

7. Preview data, configure indexes, and then click Next.

By default, Log Service enables full-text indexing. You can configure field indexes based on the logs that are collected in manual mode or automatic mode. For more information, see Configure indexes.

? Note

If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable full-text indexing and field indexing, the system uses only field indexes.

4.3.1.5.8. Collect logs from standard Docker containers

This topic describes how to deploy a Logtail container and create a Logtail configuration to collect logs from standard Docker containers.

Step 1: Deploy a Logtail container

1. Run the following command to pull the Logtail image:

docker pull registry.cn-hangzhou.aliyuncs.com/log-service/logtail

Replace *registry.cn-hangzhou.aliyuncs.com* with the actual endpoint. For more information about the endpoints of regions, see the "View the endpoint of a project" section of the Manage a project topic. If your server resides in a virtual private cloud (VPC), you must replace **registry** with **registry-vpc**.

2. Start a Logtail container.

? Note

Before you configure the parameters, you must complete one of the following configurations. Otherwise, the container text file busy error may occur when you delete other containers.

• For CentOS 7.4 and later, set fs.may_detach_mounts to 1. For more information, see Bug 1468249, Bug 1441737, and Issue 34538.

• Add --privileged to the startup parameters to grant Logtail the privileged permission. For more information, see Docker run reference.

Replace the $s[your_region_name]$, $s[your_aliyun_user_id]$, and $s[your_machine_group_user_defined_id]$ parameters in the following command with the actual values:

docker run -d -v /:/logtail_host:ro -v /var/run:/var/run --env

ALIYUN_LOGTAIL_CONFIG=/etc/ilogtail/conf/\${your_region_name}/ilogtail_config.json --env ALIYUN_LOGTAIL_USER_ID=\${your_aliyun_user_id} --env ALIYUN_LOGTAIL_USER_DEFINED_ID=\${your_machine_group_user_defined_id} registry.cn-hangzhou.aliyuncs.com/log-service/logtail

Parameter	Description
<pre>\${your_region_name}</pre>	The ID of the region where your project resides and the type of the network that your project uses.
<pre>\${your_aliyun_user_id}</pre>	The ID of your Apsara Stack tenant account. For more information, seeConfigure a user identifier.
<pre>\${your_machine_group_user_defined_id}</pre>	The custom identifier of your machine group. The identifier must be unique in the region where your project resides. For more information, see Create a custom identifier-based machine group.

? Note

- You can customize the startup parameters of the Logtail container only if the following conditions are met:
- i. The following environment variables are configured: aliyun_logTail_user_defined_id , aliyun_logTail_config , aliyun_logTail_config .
- ii. The /var/run directory of the host is mounted on the /var/run directory of the Logtail container.
- iii. The root directory of the host is mounted on the /logtail_host directory of the Logtail container.
- iv. If the the parameter is invalid : uuid=none error is returned in the /usr/local/ilogtail/ilogtail.LOG log file, you must create a file named product_uuid on the host. Then, you must enter a valid universally unique identifier (UUID) in the file, for example, 169E98C9-ABCO-4A92-B1D2-AA6239C0D261, and mount the file on the /sys/class/dmi/id/product_uuid directory of the Logtail container.

Step 2: Create a Logtail configuration

Create a Logtail configuration in the console based on your business requirements.

- To collect Docker text logs, follow the steps that you perform to collect Kubernetes text logs. For more information, see Use the Log Service console to collect container text logs in DaemonSet mode.
- To collect Docker stdout and stderr logs, follow the steps that you perform to collect Kubernetes stdout and stderr. For more information, see Use the Log Service console to collect container stdout and stderr in DaemonSet mode.
- To collect host text logs, follow the steps provided in Collect text logs.

By default, the root directory of the host is mounted on the /logtail_host directory of the Logtail container. When you configure the directory to collect logs, you must add the container directory as the prefix to the log path. For example, to collect data from the /home/logs/app_log/ directory of the host, you must set the log path to /logtail_host/home/logs/app_log/.

When you create a machine group, enter the value of the ALIYUN_LOGTAIL_USER_DEFINED_ID parameter in the **Custom Identifier** field. This value is specified in Step 1: Deploy a Logtail container.

Default fields

Docker stdout and stderr logs

The following table describes the fields that are uploaded by default for each log.

Log field	Description
time	The point in time when data is uploaded. Example: 2018-02-02T02:18:41.979147844z
source	The type of the input source. Valid values: stdout and stderr.
_image_name_	The name of the image.
_container_name_	The name of the container.
_container_ip_	The IP address of the container.

Docker file

The following table describes the fields that are uploaded by default for each log.

Log field	Description
_image_name_	The name of the image.
_container_name_	The name of the container.
_container_ip_	The IP address of the container.

Other operations

• View the status of Logtail.

You can run the docker exec \${logtail_container_id} /etc/init.d/ilogtaild status command to view the status of Logtail.

• View the version number, IP address, and startup time of Logtail.

You can run the docker exec \${logtail_container_id} cat /usr/local/ilogtail/app_info.json command to view the information of Logtail.

view the operational logs of Logial.

The operational logs of Logtail are stored in the ilogtail.LOG file in the /usr/local/ilogtail/ directory. If the log file is rotated, the generated files are compressed and stored as ilogtail.LOG.x.gz. Example:

docker exec a287de895e40 tail -n 5 /usr/local/ilogtail/ilogtail.LOG				
[2018-02-06 08:13:35.721864]	[INFO]	[8]	[build/release64/sls/ilogtail/LogtailPlugin.cpp:104]	logtail plugin Resume:start
[2018-02-06 08:13:35.722135]	[INFO]	[8]	[build/release64/sls/ilogtail/LogtailPlugin.cpp:106]	logtail plugin Resume:success
[2018-02-06 08:13:35.722149]	[INFO]	[8]	[build/release64/sls/ilogtail/EventDispatcher.cpp:369]	start add existed check point even
ts, size:0				
[2018-02-06 08:13:35.722155]	[INFO]	[8]	[build/release64/sls/ilogtail/EventDispatcher.cpp:511]	add existed check point events, si
ze:0 cache size:0 event	size:0	success	count:0	
[2018-02-06 08:13:39.725417]	[INFO]	[8]	[build/release64/sls/ilogtail/ConfigManager.cpp:3776]	check container path update flag:0
size:1				

The standard output of the container is irrelevant to this case. Ignore the following standard output:

start umount useless mount points, /shm\$|/merged\$|/mqueu\$ umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fbe2bdb95d13b1e110172ef57fe840c82155/merged: must be superuser to umount umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44beab6e69718/merged: must be superuser to unmount umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640b1e16c22dbe/merged: must be superuser to ummount xargs: umount: exited with status 255; aborting umount done start logtail ilogtail is running logtail is running

```
docker exec a287de895e40 /etc/init.d/ilogtaild stop
kill process Name: ilogtail pid: 7
kill process Name: ilogtail pid: 8
stop success
docker exec a287de895e40 /etc/init.d/ilogtaild start
ilogtail is running
```

4.3.1.5.9. Collect Kubernetes events

This topic describes how to use the eventer component to collect events from Kubernetes and send the events to Log Service.

Log Service allows you to collect events from Kubernetes by using kube-eventer. For more information about the source code for Kubernetes event collection, visit GitHub.

Create a Logtail configuration

```
Note
If you use self-managed Kubernetes, you must configure the endpoint, project, logStore, regionId, internal, accessKeyId, and
accessKeySecret parameters.
```

The following example shows the event collection configuration:

```
apiVersion: apps/v1
kind: Deployment
metadata:
 labels:
   name: kube-eventer
 name: kube-eventer
 namespace: kube-system
spec:
  replicas: 1
  selector:
   matchLabels:
     app: kube-eventer
  template:
    metadata:
     labels:
       app: kube-eventer
      annotations:
       scheduler.alpha.kubernetes.io/critical-pod: ''
    spec:
     dnsPolicy: ClusterFirstWithHostNet
      serviceAccount: kube-eventer
      containers:
       - image: registry.aliyuncs.com/acs/kube-eventer-amd64:v1.1.0-c93a835-aliyun
         name: kube-eventer
          command:
```

- "/kube-eventer" - "--source=kubernetes:https://kubernetes.default" ## .send to sls ## --sink=sls:https://{endpoint}?project={project}&logStore=k8s-event®ionId={region-id}&internal=false&accessKeyId= {accessKeyId}&accessKeySecret={accessKeySecret} - --sink=sls:https://cn-beijing.log.aliyuncs.com?project=k8s-xxxx&logStore=k8s-event®ionId=cnbeijing&internal=false&accessKeyId=xxx&accessKeySecret=xxx env: # If TZ is assigned, set the TZ value as the time zone - name: TZ value: "Asia/Shanghai" volumeMounts: - name: localtime mountPath: /etc/localtime readOnly: true - name: zoneinfo mountPath: /usr/share/zoneinfo readOnly: true resources: requests: cpu: 10m memory: 50Mi limits: cpu: 500m memory: 250Mi volumes: - name: localtime hostPath: path: /etc/localtime - name: zoneinfo hostPath: path: /usr/share/zoneinfo apiVersion: rbac.authorization.k8s.io/v1 kind: ClusterRole metadata: name: kube-eventer rules: - apiGroups: _ "" resources: - events verbs: - get - list - watch apiVersion: rbac.authorization.k8s.io/v1 kind: ClusterRoleBinding metadata: name: kube-eventer roleRef: apiGroup: rbac.authorization.k8s.io kind: ClusterRole name: kube-eventer subjects: - kind: ServiceAccount name: kube-eventer namespace: kube-system apiVersion: v1 kind: ServiceAccount metadata: name: kube-eventer namespace: kube-system

Parameter	Туре	Required	Description
endpoint	string	Yes	The Log Service endpoint. For more information, seeView the endpoint of a project.
project	string	Yes	The project in Log Service.
logStore	string	Yes	The Logstore in Log Service.
internal	string	Required for self-managed Kubernetes	If you use self-managed Kubernetes, set the value to false.
regionId	string	Required for self-managed Kubernetes	The ID of the region where the Log Service Logstore resides. For more information, see View the endpoint of a project.

accessKeyId	string	Required for self-managed Kubernetes	The AccessKey ID. We recommend that you use the AccessKey ID of a RAM user.
accessKeySecret	string	Required for self-managed Kubernetes	The AccessKey secret. We recommend that you use the AccessKey secret of a RAM user.

Sample log

The following example shows a collected sample log:

hostname: cn-hangzhou.i-*********	
level: Normal	
pod_id: 2a360760-****	
pod_name: logtail-ds-blkkr	
event_id: {	
"metadata":{	
"name":"logtail-ds-blkkr.157b7cc90de7e192",	
"namespace":"kube-system",	
"selfLink":"/api/v1/namespaces/kube-system/events/logtail-ds-blkkr.157b7cc90de7e192",	
"uid":"2aaf75ab-****",	
"resourceVersion":"6129169",	
"creationTimestamp":"2019-01-20T07:08:19Z"	
} <i>,</i>	
"involvedObject":{	
"kind":"Pod",	
"namespace":"kube-system",	
"name":"logtail-ds-blkkr",	
"uid":"2a360760-****",	
"apiVersion":"v1",	
"resourceVersion":"6129161",	
"fieldPath":"spec.containers{logtail}"	
} <i>,</i>	
"reason":"Started",	
"message":"Started container",	
"source": {	
"component":"kubelet",	
"host":"cn-hangzhou.i-*********"	
},	
"firstTimestamp":"2019-01-20T07:08:19Z",	
"lastTimestamp":"2019-01-20T07:08:19Z",	
"count":1,	
"type":"Normal",	
"eventTime":null,	
"reportingComponent":"",	
"reportingInstance":""	

Log field	Туре	Description
hostname	string	The hostname of the server where an event occurs.
level	string	The level of a log. Valid values: Normal and Warning.
pod_id	string	The unique identifier of a pod. This field is available only if the event type is related to the pod.
pod_name	string	The name of a pod. This field is available only if the event type is related to the pod.
eventid	json	The details of an event. The value of this field is a JSON string.

4.3.1.5.10. Collect container text logs

Logtail can collect and upload container text logs together with container metadata to Log Service.

Features

Logtail can collect and upload container text logs together with container metadata to Log Service. Compared with basic log file collection, Docket file collection by using Logtail has the following features:

- Allows you to specify the log path of a container without the need to manually map the log path of the container to a path on the host.
- Uses labels to specify containers for log collection.
- Uses labels to exclude containers from log collection.
- Uses environment variables to specify containers for log collection.
- Uses environment variables to exclude containers from log collection.
- Supports multi-line logs such as Java Stack logs.
- Supports automatic labeling for Docker container logs.

? Note

- The preceding labels are retrieved by running the docker inspect command. These labels are different from the labels that are specified in a Kubernetes cluster.
- The preceding environment variables are the same as the environment variables that are specified to start containers.

Limits

- Stop policy: If Logtail detects the die event on a container that is stopped, Logtail stops collecting logs from the container. If collection latency occurs, some logs that are collected before the container is stopped may be lost.
- Docker storage driver: For Docker containers, only overlay and overlay2 storage drivers are supported. If other storage drivers are used, you must mount the log directory on the on-premises host.
- Logtail running mode: Logtail must run in a container and must be deployed based on Logtail deployment solutions.

Step 1: Deploy and configure Logtail

Kubernetes

For more information about how to collect Kubernetes logs, see Install the Logtail component.

Configure Logtail on other containers

For more information about the methods used to manage other containers, such as Swarm and Mesos, see Collect standard Docker logs.

Step 2: Create a Logtail configuration for log collection

- 1. Log on to the Log Service console
- 2. In the Import Data section, select Docker File Container.
- 3. Select the project and Logstore. Then, click Next.
 - You can also click Create Now to create a project and a Logstore.

If you click the plus sign (+) next to **Data Import** below your Logstore to enter the configuration process, the system automatically skips this step. 4. Create a machine group and click **Next**.

Before you create a machine group, make sure that Logtail is installed.

Follow the on-screen instructions to install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows.

After you install Logtail, click **Complete Installation** to create a machine group. If you created a machine group, click **Use Existing Machine Groups**.

 Select the machine group in the Source Server Groups section and move the machine group to the Applied Server Groups section. Then, click Next.

() Important

If you apply a machine group immediately after you create the machine group, the heartbeat status of the machine group may be **FAIL**. This issue occurs because the machine group is not connected to Simple Log Service. To resolve the issue, you can click **Automatic Retry**. If the status is still **FAIL**, see What do I do if a Logtail machine group has no heartbeats?

Source Server Groups			Applied Server Groups	
Search by server group name	Q		Search by server group name	Q
V	d			
		>		
		<		
1 Items			0 Items	

6. Create a Logtail configuration.

The following table describes the parameters of data sources. For information about common parameters, see Configure text log collection.

Parameter	Description
Docker File	Checks whether the file that is collected from the specified data source is a Docker file.

Label Whitelist	If you want to configure the label allowlist, you must specify the LabelKey parameter. If the value of the LabelValue parameter is not empty, logs are collected from the containers whose label key-value pairs match the specified key-value pairs. If the value of the LabelValue parameter is empty, logs are collected from the containers whose label keys match the specified keys. ⑦ Note Key-value pairs are in the logical OR relation. If a label key-value pair of a container matches one of the specified key-value pairs, the logs of the container are collected. The labels that are described in this topic refer to the label information in docker inspect.
Label Blacklist	If you want to configure the label denylist, you must specify the LabelKey parameter. If the value of the LabelValue parameter is not empty, logs are not collected from the containers whose label key-value pairs match the specified key-value pairs. If the value of the LabelValue parameter is empty, logs are not collected from the containers whose label keys match the specified keys. ⑦ Note • Key-value pairs are in the logical OR relation. If a label key-value pair of a container matches one of the specified key-value pairs, the logs of the container are not collected. • The labels that are described in this topic refer to the label information in docker inspect.
Environment Variable Whitelist	If you want to configure the environment variable allowlist, you must specify the EnvKey parameter. If the value of the EnvValue parameter is not empty, logs are collected from the containers whose environment variable key-value pairs match the specified key-value pairs. If the value of the EnvValue parameter is empty, logs are collected from the containers whose environment variable keys match the specified keys. ⑦ Note • Key-value pairs are in the logical OR relation. If an environment variable key-value pair of a container matches one of the specified key-value pairs, the logs of the container are collected. • The environment variables that are described in this topic refer to the environment information configured in container startup.
Environment Variable Blacklist	If you want to configure the environment variable denylist, you must specify the EnvKey parameter. If the value of the EnvValue parameter is not empty, logs are not collected from the containers whose environment variable key-value pairs match the specified key-value pairs. If the value of the EnvValue parameter is empty, logs are not collected from the containers whose environment variable keys match the specified keys. Image: The value of the EnvValue parameter is empty, logs are not collected from the containers whose environment variable keys match the specified keys. Image: The value pairs are in the logical OR relation. If an environment variable key-value pair of a container matches one of the specified key-value pairs, the logs of the container are not collected. Image: The environment variables that are described in this topic refer to the environment information configured in container startup.

? Note

- Labels in an allowlist and a denylist are different from the labels that are defined in Kubernetes. The labels that are described in this topic refer to the label information in docker inspect.
- A namespace and a container name of a Kubernetes cluster can be mapped to a Docker label. The value of the LabelKey parameter for a namespace is io.kubernetes.pod.namespace. The value of the LabelKey parameter for a container name is io.kubernetes.container.name. For example, the namespace of the pod that you created is backend-prod and the container name is worker-server. In this case, you can set the key-value pair of an allowlist label to io.kubernetes.container.name : worker-server .Then, you can collect logs from only the worker-server container.
- In a Kubernetes cluster, we recommend that you specify only the io.kubernetes.pod.namespace and io.kubernetes.container.name labels. You can also specify the Environment Variable Whitelist parameter or the Environment Variable Blacklist parameter based on your business requirements.

7. Configure indexes for query and analysis. After you complete the settings, click Next.

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on the collected logs in manual or automatic mode. For more information, see Enable the index feature and configure indexes for a Logstore.

? Note

- If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.
- If you configure an index for a field of the Long or Double type, you cannot configure Case Sensitive or Delimiter for the field.

Configuration examples

Configure environment variables



Configure labels

Collect the logs of the containers whose container labels include io.kubernetes.container.name=nginx . The log file path is /var/log/nginx/access.log and logs are parsed in simple mode.

Unbulla : null,
"Labels": {
"annotation.io.kubernetes.container.hash": "53073f5a",
"annotation.io.kubernetes.container.restartCount": "0",
"annotation.io.kubernetes.container.terminationMessagePath": "/dev/termination-log",
"annotation.io.kubernetes.container.terminationMessagePolicy": "File",
"annotation.io.kubernetes.pod.terminationGracePeriod": "30",
"io.kubernetes.container.logpath": "/var/log/pods/ad00a078-4182 585/nginx_0.log",
"io.kubernetes.container.name": "nginx",
"io.kubernetes.docker.type": "container",
"io.kubernetes.pod.name": "example-foo-86ccd54874-r4mfh",
"io.kubernetes.pod.namespace": "default",
"io.kubernetes.pod.uid": "ad0
"io.kubernetes.sandbox.id": "
"maintainer": "NGINX Docker Maintainers <docker-maint@nginx.com>"</docker-maint@nginx.com>
},
"StopSignal": "SIGTERM"

Default fields

The following table describes the fields that are uploaded by default for each log of a common Docker container.

Log field	Description
_image_name_	The name of the image.
_container_name_	The name of the container.
_container_ip_	The IP address of the container.

The following table describes the fields that are uploaded by default for each log of a Kubernetes cluster.

Log field	Description
_image_name_	The name of the image.
_container_name_	The name of the container.
_pod_name_	The name of the pod.
namespace	The namespace where the pod resides.
_pod_uid_	The unique identifier of the pod.
_container_ip_	The IP address of the pod.

4.3.1.5.11. Collect container stdout and stderr logs

Logtail can collect and upload container standard output (stdout) and standard error (stderr) logs together with container metadata to Log Service. This topic describes how to create a Logtail configuration in the Log Service console to collect Kubernetes stdout and stderr logs.

Prerequisites

- A project and a Logstore are created. For more information, see Create a project and Create a Logstore.
- The alibaba-log-controller Helm package is installed. For more information, see Install Logtail.

Features

Logtail can collect container stdout and stderr logs, and upload the stdout and stderr logs together with container metadata to Log Service. The following features are supported by Logtail to collect container stdout and stderr logs:

- Collects stdout and stderr logs in real time.
- · Uses labels to specify containers for log collection.
- Uses labels to exclude containers from log collection.
- Uses environment variables to specify containers for log collection.
- Uses environment variables to exclude containers from log collection.
- Supports multi-line logs such as Java stack logs.
- Supports automatic labeling for Docker container logs.
- Supports automatic labeling for Kubernetes container logs.

? Note

- The preceding labels are retrieved by running the docker inspect command. These labels are different from the labels that are specified in a Kubernetes cluster.
- The preceding environment variables are the same as the environment variables that are specified to start containers.

Implementation

A Logtail container uses a UNIX domain socket to communicate with the Docker daemon. The Logtail container queries all Docker containers and finds the specified Docker containers based on the specified labels and environment variables. Logtail runs the docker logs command to collect the logs of the specified Docker containers.

When Logtail collects the stdout and stderr logs of a Docker container, Logtail periodically stores checkpoints to a checkpoint file. If Logtail is restarted, Logtail collects logs from the last checkpoint.



Limits

- Logtail version: Only Logtail V0.16.0 or later that runs on Linux can be used to collect stdout and stderr logs. For more information, see Install Logtail on a Linux server.
- Permissions: By default, Logtail uses the /var/run/docker.sock socket to access the Docker engine. You must make sure that a UNIX domain socket is available and the Logtail container has permissions to access the Docker engine.
- Multi-line logs: By default, the last multi-line log that is collected by Logtail is cached for 3 seconds. This prevents the multi-line log from being split into multiple logs due to output latency. You can set the cache time by specifying the BeginLineTimeoutMs parameter. The value of the BeginLineTimeoutMs parameter cannot be less than 1,000 ms. Otherwise, an error may occur.
- Stop policy: If Logtail detects the die event on a container that is stopped, Logtail stops collecting stdout and stderr logs from the container. If collection latency occurs, some stdout and stderr logs that are collected before the container is stopped may be lost.
- Docker logging driver: The logging driver collects stdout and stderr logs only in the JSON format from containers that use the Docker engine.
- Context: By default, logs that are collected from different containers by using a Logtail configuration are in the same context. If you want the logs of each container to be in different contexts, create a Logtail configuration for each container.
- Data processing: The collected data is contained in the content field. You can process the data by using a common processing method. For more information, see Customize Logtail plug-ins to process data.

Create a Logtail configuration

1. Log on to the Log Service console

- 2. In the Import Data section, select Docker Standard Output Container.
- 3. Select the project and Logstore. Then, click Next.

You can also click Create Now to create a project and a Logstore.

If you click the plus sign (+) next to **Data Import** below your Logstore to enter the configuration process, the system automatically skips this step.

4. Create a machine group and click **Next**.

Before you create a machine group, make sure that Logtail is installed.

Follow the on-screen instructions to install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows.

After you install Logtail, click **Complete Installation** to create a machine group. If you created a machine group, click **Use Existing Machine Groups**.

 Select the machine group in the Source Server Groups section and move the machine group to the Applied Server Groups section. Then, click Next.

() Important

If you apply a machine group immediately after you create the machine group, the heartbeat status of the machine group may be **FAIL**. This issue occurs because the machine group is not connected to Simple Log Service. To resolve the issue, you can click **Automatic Retry**. If the status is still **FAIL**, see **What do I do if a Logtail machine group has no heartbeats**?

ource Server Groups			Applied Server Groups	
Search by server group name	Q		Search by server group name	Q
	d			
		>		
		<		
			C O Harra	

6. In the Specify Data Source step, specify the data source and click Next.

Configure the parameters that are used to collect logs in the **Plug-in Config** field. Example:

(
"inputs": [
{
"type": "service_docker_stdout",
"detail": {
"Stdout": true,
"Stderr": true,
"IncludeLabel": {
"io.kubernetes.container.name": "nginx"
},
"ExcludeLabel": {
"io.kubernetes.container.name": "nginx-ingress-controller"
},
"IncludeEnv": {
"NGINX_SERVICE_PORT": "80"
},
"ExcludeEnv": {
"POD_NAMESPACE": "kube-system"
}
}
}
1
}
The type of the input source is service_docker_stdout .

Required

Parameter Type

Description

IncludeLabel	map	Yes	 The value of the IncludeLabel parameter is a map. The keys and values of the map are strings. The default value of this parameter is an empty map. This default value indicates that logs from all containers are collected. If the keys are not empty and the values are empty, logs are collected from the containers whose label keys match the specified keys. Note Key-value pairs are in the logical OR relation. If a label key-value pair of a container matches one of the specified key-value pairs, the logs of the container are collected. By default, the values in the map are strings. Logs are collected from the containers whose names match the values. If you use a regular expression to specify a value, logs are collected from the containers whose names match the regular expression. If you specify a value that starts with a caret (^) and ends with a dollar sign (\$), for example, ^(kube-system]stio-system)\$, logs are collected from a container named kube-system and a container named istio-system.
ExcludeLabel	map	No	 The value of the ExcludeLabel parameter is a map. The keys and values of the map are strings. The default value of this parameter is an empty map and indicates that logs from all containers are collected. If the keys are not empty and the values are empty, logs are not collected from the containers whose label keys match the specified keys. Note Key-value pairs are in the logical OR relation. If a label key-value pair of a container matches one of the specified key-value pairs, the logs of the container are not collected. By default, the values in the map are strings. Logs are not collected from the containers whose names match the values. If you use a regular expression to specify a value, logs are not collected from the containers with a caret (^) and ends with a dollar sign (\$), for example, ^(kube-system)\$, logs are not collected from a container named kube-system or a container named istio-system.
IncludeEnv	map	No	 The value of the IncludeEnv parameter is a map. The keys and values in the map are strings. The default value of this parameter is an empty map and indicates that logs from all containers are collected. If the keys are not empty and the values are empty, logs are collected from the containers whose environment variable keys match the specified keys. Note Key-value pairs are in the logical OR relation. If an environment variable key-value pair of a container matches one of the specified key-value pairs, the logs of the container are collected. By default, the values in the map are strings. Logs are collected from the containers whose names match the values. If you use a regular expression to specify a value, logs are collected from the containers whose names match the regular expression. If you specify a value that starts with a caret (^) and ends with a dollar sign (\$), for example, (kube-systemistio-system), logs are collected from a container named kube-system and a container named istio-system).
ExcludeEnv	map	No	 The value of the ExcludeEnv parameter is a map. The keys and values of the map are strings. The default value of this parameter is an empty map and indicates that logs from all containers are collected. If the keys are not empty and the values are empty, logs are not collected from the containers whose environment variable keys match the specified keys. Note Key-value pairs are in the logical OR relation. If an environment variable key-value pair of a container matches one of the specified key-value pairs, the logs of the container are not collected. By default, the values in the map are strings. Logs are not collected from the containers whose names match the values. If you use a regular expression to specify a value, logs are not collected from the containers whose names match the regular expression. If you specify a value that starts with a caret (^) and ends with a dollar sign (\$), for example, ^(kube-system)\$, logs are not collected from a container named kube-system or a container named istio-system.
Stdout	bool	No	Default value: true. If you set the value of this parameter to false, stdout logs are not collected.
Stderr	bool	No	Default value: true. If you set the value of this parameter to false, stderr logs are not collected.

BeginLineRegex	string	No	The regular expression that is used to match the start part in the first line of a log. The default value of this parameter is an empty string. If a line matches the specified regular expression, the line is recorded to be the first line of a new log. Otherwise, the line is recorded to be a part of the last log.
BeginLineTimeout Ms	int	No	The timeout period for the specified regular expression to match the start part in the first line of a log. Default value: 3000. Unit: ms. If no new log is generated within 3 seconds, the last log is uploaded.
BeginLineCheckLe ngth	int	No	The size of the start part in the first line of a log that matches the specified regular expression. Default value: $10 \times 1,024$. Unit: bytes. You can specify this parameter to check whether the start part in the first line of a log matches the regular expression. This improves match efficiency.
MaxLogSize	int	No	The maximum size of a log. Default value: 512 \times 1,024. Unit: bytes. If the size of a log exceeds the specified value, the log is uploaded.

? Note

• The preceding IncludeLabel and ExcludeLabel parameters are included in the label information that is retrieved by using the docker inspect command.

• A namespace and a container name of a Kubernetes cluster can be mapped to a Docker label. The value of the LabelKey parameter for a namespace is io.kubernetes.pod.namespace. The value of the LabelKey parameter for a container name is io.kubernetes.container.name. For example, the namespace of the pod that you created is backend-prod and the container name is worker-server. In this case, if you set the key-value pair of an allowlist label to io.kubernetes.container.name : worker-server , the logs of the container are collected. If you set the key-value pair of an allowlist label to io.kubernetes.container.name : worker-server , the logs of the container are collected.

• In a Kubernetes cluster, we recommend that you specify only the is.kubernetes.pod.namespace and is.kubernetes.container.name labels. You can also specify the IncludeEnv or ExcludeEnv parameter based on your business requirements.

7. Configure indexes for query and analysis. After you complete the settings, click Next.

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on the collected logs in manual or automatic mode. For more information, see Enable the index feature and configure indexes for a Logstore.

? Note

- If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.
- If you configure an index for a field of the Long or Double type, you cannot configure Case Sensitive or Delimiter for the field.

Default fields

Common Docker containers

The following table describes the fields that are uploaded by default for each log of a common Docker container.

Log field	Description
time	The point in time when data is uploaded. Example: 2018-02-02T02:18:41.979147844z
source	The type of the input source. Valid values: stdout and stderr.
_image_name_	The name of the image.
_container_name_	The name of the container.
_container_ip_	The IP address of the container.

Kubernetes

The following table describes the fields that are uploaded by default for each log of a Kubernetes cluster.

Log field	Description
time	The point in time when data is uploaded. Example: 2018-02-02T02:18:41.979147844z
source	The type of the input source. Valid values: stdout and stderr.
_image_name_	The name of the image.

_container_name_	The name of the container.
_pod_name_	The name of the pod.
namespace	The namespace where the pod resides.
_pod_uid_	The unique identifier of the pod.
_container_id_	The IP address of the pod.

Configuration examples of single-line log collection

Configure environment variables

Collect the stdout and stderr logs of the containers whose environment variables include NGINX_PORT_80_TCP_PORT=80 and exclude POD_NAMESPACE=kube-system .

Figure 1. Configuration example of environment variables

opensearch - ratioe,
"StdinOnce": false,
"Env": [
"HTTP_SVC_SERVICE_PORT_HTTP=80",
"LOG4J_APPENDER_DEMO_SPRING_BOOT_SVC_PORT= :8080",
"LOG4J_APPENDER_DEMO_SPRING_BOOT_SVC_PORT_8080_TCP_PORT=8080",
"HTTP_SVC_PORT_80_TCP_ADDR=",
"NGINX_PORT_80_TCP=tcp:// ',
"NGINX_PORT_80_TCP_PROTO=tcp",
"LOG4J_APPENDER_DEMO_SPRING_BOOT_SVC_SERVICE_PORT=8080",
"KUBERNETES_SERVICE_HOST=",
"HTTP_SVC_SERVICE_HOST="",
"HTTP_SVC_PORT_80_TCP_PROTO=tcp",
"NGINX_PORT_80_TCP_ADDR=: ",
"LOG4J_APPENDER_DEMO_SPRING_BOOT_SVC_PORT_8080_TCP_PROTO=tcp",
"KUBERNETES_SERVICE_PORT_HTTPS=443",
"KUBERNETES_PORT=tcp://: :443",
"NGINX_PORT=tcp://
"HTTP_SVC_PORT=tcp:// ::80",
"HTTP_SVC_PORT_80_TCP_PORT=80",
<u>"NGINX_SERVICE_PORT=80",</u>
"KUBERNETES_PORT_443_TCP=tcp:// :443",
"KUBERNETES_PORT_443_TCP_PROTO=tcp",
"HTTP_SVC_SERVICE_PORT=80",
"KUBERNETES_PORT_443_TCP_ADDR=17 1",
"HTTP_SVC_PORT_80_TCP=tcp:// :80",

The following script shows the configurations of the environment variables:

Configure labels

Collect the stdout and stderr logs of the containers whose labels include io.kubernetes.container.name=nginx and exclude type=pre .

Figure 2. Configuration example of labels

"OnBuild": null,
"Labels": {
"annotation.io.kubernetes.container.hash": "53073f5a",
"annotation.io.kubernetes.container.restartCount": "0",
"annotation.io.kubernetes.container.terminationMessagePath": "/dev/termination-log",
"annotation.io.kubernetes.container.terminationMessagePolicy": "File",
"annotation.io.kubernetes.pod.terminationGracePeriod": "30",
"io.kubernetes.container.logpath": "/var/log/pods/ad00a07885/nginx_0.log",
"io.kubernetes.docker.type": "container",
"io.kubernetes.pod.name": "example-foo-86ccd54874-r4mfh",
"io.kubernetes.pod.namespace": "default",
"io.kubernetes.pod.uid": "ad00a07
"io.kubernetes.sandbox.id": "5216 a8d0b6891dfa6da112969",
"maintainer": "NGINX Docker Maintainers <docker-maint@nginx.com>"</docker-maint@nginx.com>
},
"StopSignal": "SIGTERM"

The following script shows the label configurations:



Configuration examples of multi-line log collection

Before you can collect Java exception stack logs, you must configure multi-line log collection. The following section describes how to collect stdout and stderr logs of standard Java applications.

Sample log

```
2018-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : service start
2018-02-03 14:18:41.969 ERROR [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : java.lang.NullPointerException
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:199)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96)
...
2018-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : service start done
```

Log collection configuration

Collect the logs of the containers whose labels include app=monitor and the specified first bytes of a line is of a fixed-format date type. To improve match efficiency, only the first 10 bytes of each line are checked.



Data processing examples

Logtail can process the collected Docker standard output. For more information, see Common data processing methods.

Collect the logs of the containers whose labels include app=monitor and the specified first bytes of a line is of a fixed-format data type. To improve
match efficiency, only the first 10 bytes of each line are checked. Regular expressions are used to parse logs into the values of the time, level,
module, thread, and message. The following script shows the configurations of log collection and data processing:

User Guide-Log Service

"inputs": [
{
"detail": {
"BeginLineCheckLength": 10,
"BeginLineRegex": "\\d+-\\d+-\\d+.*",
"IncludeLabel": {
"app": "monitor"
}
},
"type": "service_docker_stdout"
}
1,
"processors": [
(
"type": "processor_regex",
"detail": {
"SourceKey": "content",
$\label{eq:started} \label{eq:started} eq:s$
"Keys": [
"time",
"level",
"module",
"thread",
"message"
],
"NoKeyError": true,
"NoMatchError": true,
"KeepSource": false
}
}
]

The collected log 2018-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : service start done is processed, as shown in the following script:

__tag__:_hostname__:logtail-dfgef _container_name_:monitor _image_name__registry.cn-hangzhou.aliyuncs.xxxxxxxxxxxxx _namespace_:default _pod_name_imonitor-6f54bd5d74-rtzc7 _pod_uid_:?f012b72-04c7-11e8-84aa-00163f00c369 _source_istdout _time_:2018-02-0202114;18:41.979147844Z time:2018-02-02 02:18:41.968 level:INFO module:spring-cloud-monitor thread:nio-8080-exec-4 classic.g.s.web.controller.DemoController message:service start done

Collect the JSON logs of the containers whose labels include app=monitor. The following script shows the configurations of log collection and data processing:



4.3.1.5.12. Collect standard Docker logs

This topic describes how to use Logtail to collect standard Docker logs and upload these logs together with the container metadata to Log Service.

Procedure

}

Figure 1. Procedure



- 1. Deploy a Logtail container.
- 2. Configure a Logtail server group.

Create a server group with a custom ID in the Log Service console. The container cluster can automatically scale up or down based on data traffic. 3. Create a Logical configuration.

3. Create a Logial configuration.

Create a Logtail configuration in the Log Service console. The Logtail configuration process is completed in the Log Service console. No local configuration is needed.

Deploy a Logtail container

1. Run the following command to pull the Logtail image.

docker pull registry.cn-hangzhou.aliyuncs.com/log-service/logtail

2. Start a Logtail container.

Set the \$(your_region_name) , \$(your_aliyun_user_id) , and \$(your_machine_group_user_defined_id) parameters in the startup template.

docker run -d -v /:/logtail_host:ro -v /var/run:/var/run --env

ALIYUN_LOGTAIL_CONFIG=/etc/ilogtail/conf/\${your_region_name}/ilogtail_config.json

--env ALIYUN_LOGTAIL_USER_ID=\${your_aliyun_user_id} --env

ALIYUN_LOGTAIL_USER_DEFINED_ID=\${your_machine_group_user_defined_id} registry.cn-hangzhou.aliyuncs.com/log-service/logtail

() Important

Before you set the parameters, you must complete one of the following configurations. Otherwise, the container text file busy error may occur when you delete another container.

- For CentOS 7.4 and later versions, set fs.may_detach_mounts to 1. For more information, see Bug 1468249, Bug 1441737, and Issue 34538.
- Grant the privileged permission to Logtail by adding the --privileged flag to the startup parameters. For more information, see Docker run reference.

Parameter	Description
<pre>\${your_region_name}</pre>	The region of the project. For more information, seeManage a project.
<pre>\$(your_aliyun_user_id)</pre>	The user ID. Set this parameter to the ID of your Alibaba Cloud account, which is a string. For information about how to view the ID, see Step 1 in Configure a user identifier.
<pre>\${your_machine_group_user_defined_id}</pre>	The custom ID of your server group. For information about how to set the custom ID, see Step 1 inCreate a custom identifier-based machine group.

After you set the parameters, run the following command to start the Logtail container.

docker run -d -v /:/logtail_host:ro -v /var/run:/var/run

--env ALIYUN_LOGTAIL_CONFIG=/etc/ilogtail/conf/cn_hangzhou/ilogtail_config.json --env

ALIYUN_LOGTAIL_USER_ID=1654218******--env ALIYUN_LOGTAIL_USER_DEFINED_ID=log-docker-demo registry.cn-hangzhou.aliyuncs.com/logservice/logtail
() Important

- You can customize the startup parameters of the Logtail container if the following conditions are met:
 - The following environment variables exist before you start the Logtail container: ALIYUN_LOGTAIL_USER_DEFINED_ID , ALIYUN_LOGTAIL_USER_ID , and ALIYUN_LOGTAIL_USER_ID .
 - The /var/run directory is mounted on the /var/run directory of the Logtail container.
 - To collect container standard output, container logs, or host files, you must mount the root directory on the /logtail_host directory of the Logtail container.
 - If an error showing The parameter is invalid : uuid=none occurs in the /usr/local/ilogtail/ilogtail.Log Logtail log file, create a file named product_uuid on the host. Add a valid UUID such as 169E98C9-ABC0-4A92-B1D2-AA6239C0D261 to the file, and mount the file on the /sys/class/dmi/id/product_uuid directory of the Logtail container.

Configure a Logtail server group

- 1. Log on to the Log Service console
- 2. Click a project name.
- 3. In the left-side navigation pane, click the Server Groups icon to show the server group list.
- 4. Click the icon next to Server Groups, and then select Create Server Group.
- You can also create a server group when you import data to Log Service.
- 5. In the dialog box that appears, select **Custom ID** from the Identifier drop-down list. Enter the value of <u>ALIYUN_LOGTAIL_USER_DEFINED_ID</u> set in the previous step in the **Custom Identifier** field.

Click OK. One minute later, click the name of the server group in the **Server Groups** list. On the **Server Group Settings** page that appears, you can view the heartbeat status of the Logtail container. For more information, see View the status of a server group.

Create a Logtail configuration

Create a Logtail configuration in the console.

- For more information about Docker logs, see Collect container text logs.
- For more information about Docker standard output, see Collect container stdout and stderr logs.
- Host text logs
- The root directory of a host is mounted on the /logtail_host directory of the Logtail container by default. You must add the /logtail_host prefix to the log path. For example, if you want to collect data from the /home/logs/app_log/ directory of the host, you must set the log path as /logtail_host/home/logs/app_log/ .

What to do next

- View the status of the Logtail container.
 - You can run the docker exec {logtail_container_id} /etc/init.d/ilogtaild status command to view the status of Logtail.
- View the version number, IP address, and startup time of Logtail.

You can run the docker exec {{logtail_container_id} cat /usr/local/ilogtail/app_info.json command to view Logtail information.

• View the operations logs of Logtail.

The operations logs of Logtail are stored in the ilogtail.LOG file in the /usr/local/ilogtail/ directory. If the log file is rotated and compressed, it is stored as a file named ilogtail.LOG.x.gz .

For example:

[root@iZbp17enxc2us3624wexh2Z	ilogtail]#	docker	exec a287de895e40 tail -n 5 /usr/local/ilogtail/ilogtail	LOG
[2018-02-06 08:13:35.721864]	[INFO]	[8]	[build/release64/sls/ilogtail/LogtailPlugin.cpp:104]	logtail plugin Resume:start
[2018-02-06 08:13:35.722135]	[INFO]	[8]	[build/release64/sls/ilogtail/LogtailPlugin.cpp:106]	logtail plugin Resume:success
[2018-02-06 08:13:35.722149]	[INFO]	[8]	[build/release64/sls/ilogtail/EventDispatcher.cpp:369]	start add existed check point even
ts, size:0				
[2018-02-06 08:13:35.722155]	[INFO]	[8]	[build/release64/sls/ilogtail/EventDispatcher.cpp:511]	add existed check point events, si
ze:0 cache size:0 event	size:0	success	count:0	
[2018-02-06 08:13:39.725417]	[INFO]	[8]	[build/release64/sls/ilogtail/ConfigManager.cpp:3776]	check container path update flag:0
size:1				

Ignore the following standard output:

start umount useless mount points, /shm\$!/merged\$!/mqueus\$ umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fbe2bdb95d13ble110172ef57fe840c82155/merged: must be superuser to unmount umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44beab6e69718/merged: must be superuser to unmount umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640ble16c22dbe/merged: must be superuser to unmount ... xargs: umount: exited with status 255; aborting umount done start logtail ilogtail is running logtail is running

To restart Logtail, use the following sample code:

[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 /etc/init.d/ilogtaild stop
kill process Name: ilogtail pid: 7
kill process Name: ilogtail pid: 8
stop success
[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 /etc/init.d/ilogtaild start
ilogtail is running

4.3.1.6. Custom plug-ins

Background information

Log Service allows you to collect text logs and system logs by using Logtail. Logtail supports connections with multiple data sources, such as HTTP or MySQL query results and MySQL binary logs.

You can collect HTTP request data and upload the processing results to Log Service in real time to check service availability and perform continuous availability monitoring. You can configure MySQL query results as the data source, and then synchronize incremental data based on custom IDs or time. You can also configure an SQL data source to synchronize MySQL binary logs, subscribe to database changes, and query or analyze logs in real time.

() Note This feature supports only Logtail V0.16.0 or later that runs on Linux. For more information about Logtail versions and version updates, see Install Logtail on a Linux server.

Configuration process

- 1. Configure a collection method for a data source.
- Configure collection methods based on different data sources.
- 2. Configure a data processing method.

Logtail provides multiple processing methods for binary logs, MySQL query results, NGINX monitoring data, and HTTP input sources. You can configure multiple processing methods for a single input source. Each input source supports all processing methods. Logtail runs the configured processing methods in sequence.

For more information, see Customize Logtail plug-ins to process data.

- 3. Apply the configurations to the specified machine group.
- Apply the log collection configurations and processing configurations to the specified machine group. Then, Logtail automatically pulls the configurations and starts to collect logs.

4.3.1.6.1. Collect MySQL binary logs

This topic describes how to create a Logtail configuration in the Log Service console to collect MySQL binary logs.

Prerequisites

Logtail is installed on the server from which you want to collect MySQL binary logs. For more information, see Install Logtail on a Linux server or Install Logtail on a Windows server.

⑦ Note Linux servers support Logtail V0.16.0 or later. Windows servers support Logtail V1.0.0.8 or later.

How it works

Logtail acts as a secondary MySQL node to communicate with the primary MySQL node. The following list describes the communication process:

- 1. Logtail acts as a secondary MySQL node and sends a dump request to the primary MySQL node.
- 2. After the primary MySQL node receives the dump request, the node sends binary logs to Logtail in real time.
- Logtail performs operations such as event parsing, event filtering, and data parsing on binary logs. Then, Logtail uploads the parsed data to Log Service.





Features

- Binary logs can be incrementally collected. This way, you can collect data related to the update operations that are performed on your databases. MySQL databases such as ApsaraDB RDS for MySQL are supported.
- Multiple methods are provided to filter data in databases, such as regular expressions.
- You can specify the positions of binary log files.
- Checkpoints are used to synchronize data storage status.

Limits

• Binary logs of MySQL 8.0 or later cannot be collected.

• The binary logging feature must be enabled for your MySQL database, and the binlog_format parameter must be set to ROW for the database. By default, the feature is enabled for an RDS database.

 $\circ~$ You can run the following command to check whether the binary logging feature is enabled:

show variables like "log_bin";

In this example, the following output is returned:

```
+----+

Variable_name | Value |

+----+

| log_bin | ON |

+----+

1 row in set (0.02 sec)
```

• Run the following command to view the format of binary logs:

how variables like "binlog_format";

In this example, the following output is returned:

```
+----+
Variable_name | Value |
+----+
1 binlog_format | ROW |
+---++
1 row in set (0.03 sec)
```

• The ID of the secondary MySQL node whose role Logtail assumes must be unique on the primary MySQL node.

• Limits on RDS databases:

- Logtail cannot be installed on a server where an RDS instance resides. You must install Logtail on a server that can communicate with the RDS instance.
- · You cannot collect binary logs from a secondary RDS database. You must configure your primary RDS database to collect binary logs.

Scenarios

The MySQL binary logging feature applies to scenarios in which you need to synchronize large amounts of data and require high performance.

- · Query and analyze the incremental data of databases in real time.
- Audit the operations that are performed on databases.
- Use Log Service to query and analyze database updates, visualize query and analysis results, transform data for stream computing, export log data to MaxCompute for offline computing, and export log data to Object Storage Service (OSS) for long-term storage.

Usage notes

We recommend that you increase resource limits on Logtail to accommodate traffic surges and prevent data risks. If the limits are exceeded, Logtail may be forced to restart.

You can modify the related parameters in the /usr/local/ilogtail/ilogtail_config.json file. For more information, see Configure the startup parameters of Logtail.

The following example shows how to increase the limit on CPU utilization to two cores and the limit on memory usage to 2,048 MB:

{ "cpu_usage_limit":2, "mem_usage_limit":2048,

```
}
```

Data reliability

We recommend that you enable the global transaction identifier (GTID) feature on your MySQL server and upgrade Logtail to V0.16.15 or later. This prevents data from being repeatedly collected after a primary/secondary switchover is triggered on your database and ensures data reliability.

 Incomplete data collection: If the network between Logtail and your MySQL server is disconnected for a long period of time, some data may not be collected.

If the network between Logtail and your primary MySQL node is disconnected, the primary node still generates binary logs and deletes expired binary logs. After the network connection is re-established and your primary MySQL node and Logtail are reconnected, Logtail uses a checkpoint to request binary log data from the primary MySQL node. However, if the network is disconnected for a long period of time, the data that is generated after the checkpoint may be deleted. In this case, the recovery mechanism is triggered. The recovery mechanism identifies the most recent binary log file position from which Logtail resumes collection on the primary MySQL node. The data that is generated between the checkpoint and the most recent binary log file position is not collected. This leads to incomplete data collection.

Repeated data collection: If a primary/secondary switchover is triggered when the sequence numbers of binary logs are inconsistent between your
primary MySQL node and secondary MySQL node, binary logs may be repeatedly collected.

If you configure primary/secondary synchronization for MySQL, the primary node automatically synchronizes the binary logs to the secondary node. The secondary node stores the logs to local binary log files. If the sequence numbers are inconsistent between the primary and secondary nodes and a primary/secondary switchover is triggered, logs may be repeatedly collected. This issue occurs because the checkpoint mechanism identifies checkpoints based on the names of binary log files and the offsets of the files.

For example, a data block is in the checkpoint range from (binlog.100, 4) to (binlog.105, 4) on the primary MySQL node, and in the checkpoint range from (binlog.1000, 4) to (binlog.1005, 4) on the secondary MySQL node. Logtail has obtained the data from the primary node and updated the local checkpoint to (binlog.105, 4). If a primary/secondary switchover is triggered and no error occurs, Logtail continues to collect binary logs from the new primary node based on the local checkpoint (binlog.105, 4). However, the sequence number of the data that is in the checkpoint range from (binlog.1000, 4) to (binlog.1005, 4) on the new primary node is greater than the sequence number of the data that is that is requested by Logtail. The new primary node returns all data in the range to Logtail. This leads to repeated data collection.

Procedure

1. Log on to the Simple Log Service console

- 2. In the Import Data section, select MySQL BinLog Plug-in.
- 3. Select the project and Logstore. Then, click Next.
 - You can also click Create Now to create a project and a Logstore.

If you click the plus sign (+) next to **Data Import** below your Logstore to enter the configuration process, the system automatically skips this step. 4. Create a machine group and click **Next**.

Before you create a machine group, make sure that Logtail is installed.

Follow the on-screen instructions to install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows.

After you install Logtail, click **Complete Installation** to create a machine group. If you created a machine group, click **Use Existing Machine Groups**.

5. Select the machine group in the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

! Important

If you apply a machine group immediately after you create the machine group, the heartbeat status of the machine group may be **FAIL**. This issue occurs because the machine group is not connected to Simple Log Service. To resolve the issue, you can click **Automatic Retry**. If the status is still **FAIL**, see **What** do I do if a Logtail machine group has no heartbeats?

Source Server Groups		A	pplied Server Groups	
Search by server group name	Q	>	Search by server group name	Q
1 Items			0 Items	

- 6. In the Specify Data Source step, configure the Config Name and Plug-in Config parameters.
 - A template is provided for the Plug-in Config parameter. You can configure the inputs and processors parameters in the template.
 - inputs: specifies the collection configurations. This parameter is required. Configure the inputs parameter based on your data source.

?	Note
You	I can configure only one type of data source in the inputs field.

• processors: specifies the processing method. This parameter is optional. You can configure one or more processing methods in the processors field. For more information, see Customize Logtail plug-ins to process data.

1					
"inputs	": [
{					
	"type": "se	ervice_canal",			
	"detail":				
	"Host":	"*********	mysql.rds.aliyu	incs.com",	
	"Port":	3306,			
	"User"	: "root",			
	"Server	ID" : 56321,			
	"Passwo	ord": "******",			
	"Includ	deTables": [
	"us	ser info*"			
	1.				
	"Exclud	deTables": (
	",	\\.\\S+ inner"			
	1.				
	"TextTo	String" • true			
	"Enable	DDL" : true			
	1				
1	1				
1					
1					
1					
Parame	ter	Туре	Required		Description
		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,			

type	string	Yes	The type of the data source. Set the value to service_canal .
Host	string	No	The IP address of the primary MySQL server. Default value: 127.0.0.1 .
Port	int	No	The port that is used to access the database of the primary MySQL server. Default value: 3306 .
User	string	No	The username of the account that is used to log on to the database. Default value root . Make sure that the user is granted the read permissions on the database and the REPLICATION permission. Example: CREATE USER canal IDENTIFIED BY 'canal'; GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'canal'@'%'; GRANT ALL PRIVILEGES ON *.* TO 'canal'@'%'; FLUSH PRIVILEGES;
Password	string	No	The password of the account that is used to log on to the database. By default, this parameter is left empty. If you have high requirements for data security, we recommend that you set the username and the password to xxx. After your configurations are synchronized to the on-premises server, find the Password parameter in the <i>usrylloalilyide</i> _log_config.joon file and change the value. For more information, see Modify the Logtail configuration on the Logtail server. Note If you modify this parameter in the console, the on-premises configuration is overwritten after the modification is synchronized to the on-premises server.
ServerID	int	No	The ID of the secondary MySQL node whose role Logtail assumes. Default value: 125 .
IncludeTables	String array	Yes	The names of the tables from which data is collected. Each name must include the name of the database to which a table belongs and the name of the table. Example: test_db.test_table. You can specify a regular expression for this parameter. Logtail collects data only from tables whose names match the regular expression specified by the IncludeTables parameter. To collect data from all tables of a database, set the IncludeTables parameter to .**. O Note To implement exact match, add ^ to the beginning of a regular expression and\$ to the end. Example: ^test_db\\.test_table\$.
ExcludeTables	String array	No	The names of the tables from which data is not collected. Each name must include the name of the database to which a table belongs and the name of the table. Example: test_db.test_table. You can specify a regular expression for this parameter. If a table meets one of the conditions that are specified by the ExcludeTables parameter, data from the table is not collected. If you do not configure this parameter, data from all tables is collected. ⑦ Note To implement exact match, add ^ to the beginning of a regular expression and\$ to the end. Example: ^test_db\.test_table\$.

StartBinName	string	Νο	The name of the binary log file from which Logtail starts to collect data for the first time. If you do not configure this parameter, Logtail starts to collect data from the current time. If you want Logtail to collect data from a specified position of a binary log file, set the StartBinName parameter to the name of the binary log file and set theStartBinlogPos parameter to the offset of the file. For example, you can set the StartBinName parameter to "mysql-bin". 000063" and the StartBinlogPos parameter to 0. show binary logs; Example:
StartBinlogPos	int	No	The offset of the binary log file from which Logtail starts to collect data for the first time. Default value: ${f 0}.$
EnableGTID	bool	No	Specifies whether to add GTID. For more information, see GTID. Default value: true . If you set the value to false , no GTIDs are added to the data that is uploaded to Log Service.
EnableInsert	bool	No	Specifies whether to collect the data on INSERT events. Default value: true . If you set the value to false , Logtail does not collect the data on INSERT events.
EnableUpdate	bool	No	Specifies whether to collect the data on UPDATE events. Default value: true . If you set the value to false , Logtail does not collect the data on UPDATE events.
EnableDelete	bool	No	Specifies whether to collect the data on DELETE events. Default value: true . If you set the value to false , Logtail does not collect the data on DELETE events.
EnableDDL	bool	No	Specifies whether to collect the data on data definition language (DDL) events. Default value: false. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates the data on DDL events. This value indicates that Logtail does not collect the data on DDL events. This value indicates the data on DDL events. This va
Charset	string	No	The encoding format. Default value: utf-8 .
TextToString	bool	No	Specifies whether to convert the data of the text type to the string type. Default value false . This value indicates that the data type is not converted.
PackValues	bool	No	Specifies whether to pack event data in the JSON format. Default value : false. This value indicates that Logtail does not pack event data. If you set the value to true, Logtail packs event data into the data and old_data fields in the JSON format. Theold_data field is available only for ROW_UPDATE events. For example, a table contains three columns named c1, c2, and c3. If you set the value to true, Logtail packs all data in the c1, c2, and c3 columns into the data field whose values are in the {"c1":"", "c2": "", "c3": ""} format.
EnableEventMeta	bool	No	Specifies whether to collect the metadata of events. Default value: false . This value indicates that Logtail does not collect the metadata of events. The metadata of binary log events includes event_time, event_log_position, event_size, and event_server_id. Note This parameter is available only for Logtail V0.16.21 and later.

7. Configure indexes for query and analysis. After you complete the settings, click Next.

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on the collected logs in manual or automatic mode. For more information, see Enable the index feature and configure indexes for a Logstore.

? Note

- If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field
- indexing, the system uses only field indexes
- If you configure an index for a field of the Long or Double type, you cannot configure Case Sensitive or Delimiter for the field.

After the Logtail configuration is delivered to the Logtail server, Logtail immediately collects and sends data to Log Service when changes are made to your database.

? Note

By default, Logtail collects the incremental data of binary logs

Modify the Logtail configuration on the Logtail server

If you did not enter real information for parameters such as Host, User, and Password in the **Plug-in Config** field when you created the Logtail configuration, you can modify the parameters after the Logtail configuration is delivered to the Logtail server.

- 1. Log on to the server where Logtail is installed.
- 2. Find the service_canal keyword in the /usr/local/ilogtail/user_log_config.json file and modify parameters such as Host, User, and Password.
- 3. Run the following command to restart Logtail:

sudo /etc/init.d/ilogtaild stop; sudo /etc/init.d/ilogtaild start

What to do next

After Logtail collects and sends MySQL binary logs to Log Service, you can view the logs in the Log Service console. For example, after you perform the INSERT, UPDATE, and DELETE operations on the SpecialAlarm table of the user_info database, Logtail collects and sends binary logs to Log Service. The following list describes the schema of the table, the operations that are performed on the table, and the collected logs.

• Table schema

CREATE TABLE `SpecialAlarm` (`id` int(11) unsigned NOT NULL AUTO_INCREMENT, `time` datetime NOT NULL, `alarmtype` varchar(64) NOT NULL, `ip` varchar(16) NOT NULL, `count` int(11) unsigned NOT NULL, PRIMARY KEY (`id`), KEY `ilme` (`time`) USING BTREE, KEY `alarmtype` (`alarmtype`) USING BTREE) ENGINE=MyISAM AUTO_INCREMENT=1;

Database operations

Perform the INSERT, DELETE, and UPDATE operations.

insert into specialalarm (`time`, `alarmType`, `ip`, `count`) values(now(), "NO_ALARM", " 203.0.**.***", 55); delete from specialalarm where id = 4829235 ; update specialalarm set ip = " 203.0.***.**" where id = "4829234";

Create an index for zc.specialalarm .

ALTER TABLE `zc`.`specialalarm` ADD INDEX `time_index` (`time` ASC);

Collected logs

You can view the logs that are collected for each operation on the Search & Analysis page of the Logstore that is specified in the Logtail configuration. Examples:

INSERT statement

```
__source__: 203.0.**.**
__tag_:__hostname__: iZbp145dd9fccu*****
__topic__:
_db_: zc
__event_: row_insert
__gtid_: 7d2ea78d-b631-11e7-8afb-00163e0eef52:536
__host_: *********.mysql.rds.aliyuncs.com
_id_: 113
__table_: specialalarm
alarmtype: NO_ALARM
count: 55
id: 4829235
ip: 203.0***.***
time: 2017-11-01 12:31:41
```

• DELETE statement

__source_: 10.30.**.**
__tag_:_hostname_: iZbp145dd9fccu****
__topic_:
db: zc
__event_: row_delete
__gtid_: 7d2ea78d-b631-11e7-8afb-00163e0eef52:537
host: *********.mysql.rds.aliyuncs.com
__id_: 114
__table_: specialalarm
alarmtype: NO_ALARM
count: 55
id: 4829235
ip: 10.10.**.***
time: 2017-11-01 12:31:41

• UPDATE statement

__source_: 203.0**.**
__tag_:_hostname_: iZbp145dd9fccu****
__topic_:
db: zc
gtid: 7d2ea78d-b631-11e7-8afb-00163e0eef52:538
host: ********.mysql.rds.aliyuncs.com
id: 115
_old_alarmtype: NO_ALARM
_old_count: 55
_old_id: 4829234
_old_ip: 203.0.113.1
_old_time: 2017-10-31 12:04:54
__table_: specialalarm
alarmtype: NO_ALARM
count: 55
id: 4829234
ip: 203.0.***.*t
time: 2017-10-31 12:04:54

• DDL statement

__source__: 203.0.**.**
__tag_:_hostname__: iZbp145dd9fccu****
__topic__:
db: zc
__event_: row_update
__gtid_: 7d2ea78d-b631-11e7-8afb-00163e0eef52:539
host: ********.mysql.rds.aliyuncs.com
ErrorCode: 0
ExecutionTime: 0
Query: ALTER TABLE `zc`.`specialalarm`
ADD INDEX `time_index` (`time` ASC)
StatusVars:

Field	Description
host	The hostname of the database.
db	The name of the database.
table	The name of the table.
event	The type of the event.
id	The auto-increment ID. IDs start from 0 and increment by 1 each time data on a binary log event is collected.
gtid	The GTID.
filename	The name of the binary log file.
offset	The offset of the binary log file. The value is updated only when a COMMIT operation is performed.

4.3.1.6.2. Collect MySQL query results

This topic describes how to create a Logtail configuration in the Log Service console to collect MySQL query results.

Prerequisites

Logtail is installed on the server from which you want to collect MySQL query results. For more information, see Install Logtail on a Linux server or Install Logtail on a Windows server.

? Note

Linux servers support Logtail V0.16.0 or later. Windows servers support Logtail V1.0.0.8 or later.

Implementation

Logtail executes the SELECT statement that is specified in a Logtail configuration on a regular basis, and then uploads the query results to Log Service. After Logtail obtains query results, Logtail saves the value of the CheckPoint field in the results to the Logtail server. The next time Logtail executes the SELECT statement, Logtail adds the value of the CheckPoint field to the SELECT statement. This way, Logtail can collect incremental data.



Features

- MySQL databases are supported.
- You can configure paged query settings.
- You can specify time zones
- You can specify timeout periods.
- The values of the CheckPoint field can be saved.
- SSL is supported.
- You can specify the maximum size of data that can be collected at a time.

Scenarios

- Collect incremental data based on marks such as an auto-increment ID or a point in time.
- Synchronize data based on filter conditions.

Procedure

The following procedure describes how to synchronize incremental data from a MySQL database to Log Service. In this procedure, the logtail.VersionOs field is synchronized every 10 seconds and the value of the count parameter in this field is greater than 0. The value of the initial checkpoint is 2017-09-25 11:00:00. Logs are paginated and each page contains 100 logs. The checkpoint of each page is saved. The procedure includes the following steps:

- 1. Log on to the Simple Log Service console
- 2. Select a data source.

In the Import Data section, select MySQL Query Result - Plug-in.

- Select the project and Logstore. Then, click Next.
 - You can also click Create Now to create a project and a Logstore.

If you click the plus sign (+) next to **Data Import** below your Logstore to enter the configuration process, the system automatically skips this step. 4. Create a machine group and click **Next**.

Before you create a machine group, make sure that Logtail is installed.

Follow the on-screen instructions to install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows.

After you install Logtail, click **Complete Installation** to create a machine group. If you created a machine group, click **Use Existing Machine Groups**.

5. Select the machine group in the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

() Important

If you apply a machine group immediately after you create the machine group, the heartbeat status of the machine group may be **FAIL**. This issue occurs because the machine group is not connected to Simple Log Service. To resolve the issue, you can click **Automatic Retry**. If the status is still **FAIL**, see What do I do if a Logtail machine group has no heartbeats?

Search by server group name Q	Search by server group name	Q
g hat group of the man conversion of conversed	≻ <	
	> <	
	> <	
	➤	
	<	

- 6. In the Specify Data Source step, configure the Config Name and Plug-in Config parameters.
 - In the Plug-in Config field, modify the parameter settings in the default configuration template based on your business requirements.
 - inputs : specifies the collection configurations. This parameter is required. processors : specifies the processing method. This parameter is optional. You must specify statements to collect data based on your data source. For more information, see Customize Logtail plug-ins to process data.

? Note

If you have high requirements for data security, we recommend that you set the username and the password to xxx. After your configurations are synchronized to the on-premises server, find the User and Password parameters in the /usr/local/ilogtail/user_log_config.json file and change the values.

The following example shows the configurations:

```
{
   "inputs": [
     {
        "type": "service_mysql",
        "detail": {
          'detail': {
    "Address": "***********.mysql.rds.aliyuncs.com",
    "User: "*****",
    "Password": "******",
    "DataBase": "****",
          "Limit": true,
           "PageSize": 100,
           "StateMent": "select * from db.VersionOs where time > ?",
           "CheckPoint": true,
          "CheckPointColumn": "time",
"CheckPointStart": "2018-01-01 00:00:00",
           "CheckPointSavePerPage": true,
           "CheckPointColumnType": "time",
           "IntervalMs": 60000
        }
     }
   ]
}
```

Parameter	Туре	Required	Description
type	string	Yes	The type of the data source. Set the value to service_mysql .
Address	string	No	The address of the MySQL database. Default value:127.0.0.1:3306.
User	string	No	The username of the account that you use to log on to the MySQL database. Default value: root .

Password	string	No	The password of the account that you use to log on to the MySQL database. By default, this parameter is left empty. If you have high requirements for data security, we recommend that you set the username and the password to xxx. After your configurations are synchronized to the on-premises server, find the User and Password parameters in the /usr/local/ilogtail/user_log_config.json file and change the values. (?) Note If you modify this parameter in the console, the on-premises configuration is overwritten after the modification is synchronized to the on-premises server.
DialTimeOutMs	int	No	The timeout period for connections to the MySQL database. Unit: milliseconds. Default value: 5000.
ReadTimeOutMs	int	No	The timeout period for data reads from the MySQL database. Unit: milliseconds. Default value: 5000.
StateMent	string	No	The SQL statement. If you set the CheckPoint parameter to true , you must include the CheckPointColumn parameter in the WHERE clause of the SQL statement that you specified for the StateMent parameter. You must also set the CheckPointColumn parameter to ? . For example, if you set the CheckPointColumn parameter to id , you must specify the value of the StateMent parameter in the SELECT * from where id > ? format.
Limit	bool	No	Specifies whether to use a LIMIT clause to paginate query results. Default value: false . This value indicates that query results are not paginated. We recommend that you set the Limit parameter to true , a LIMIT clause is automatically appended to the SQL statement that you specified for the StateMent parameter when Logtail executes the SQL statement.
PageSize	int	No	The maximum number of logs to return on each page. If you set th d.imit parameter to true , you must configure this parameter.
MaxSyncSize	int	No	The maximum number of logs that can be synchronized at a time. Default value: ${\bf 0}$. This value indicates that no limit is placed on the size of data that can be synchronized at a time.
CheckPoint	bool	No	Specifies whether to use checkpoints during data collection. Default value: false . This value indicates that checkpoints are not used during data collection.
CheckPointColumn	string	No	The name of the checkpoint column. If you set the CheckPoint parameter to true , you must configure this parameter.
CheckPointColumnTyp e	string	No	The type of the checkpoint column. Valid values: int and time. If you set this parameter to int, the values in the checkpoint column are stored as 64-bit integers. If you set this parameter to time, the values in the checkpoint column can be of the date, time, or datetime type that is supported by MySQL. If you set the CheckPoint parameter to true , you must configure this parameter.
CheckPointStart	string	No	The initial value of the checkpoint. If you set the CheckPoint parameter to true , you must configure this parameter.
CheckPointSavePerPa ge	bool	No	If you set this parameter to true , a checkpoint is saved after each pagination. If you set this parameter to false , a checkpoint is saved after each synchronization.
IntervalMs	int	Yes	The synchronization interval. Unit: milliseconds.

7. Configure indexes for query and analysis. After you complete the settings, click **Next**.

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on the collected logs in manual or automatic mode. For more information, see Enable the index feature and configure indexes for a Logstore.

? Note

 If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.

• If you configure an index for a field of the Long or Double type, you cannot configure Case Sensitive or Delimiter for the field.

Modify the configurations on the server where Logtail is installed

If you did not enter real information for parameters such as Address, User, and Password in the **Plug-in Config** field when you created the Logtail configuration, you can modify the parameters after the Logtail configuration is delivered to the Logtail server.

- 1. Log on to the server where Logtail is installed.
- Find the service_mysql keyword in the /usr/local/ilogtail/user_log_config.json file and modify parameters such as Address, User, and Password.
 Run the following command to restart Logtail:

sudo /etc/init.d/ilogtaild stop; sudo /etc/init.d/ilogtaild start

Example

After Logtail collects and sends MySQL query results to Log Service, you can view the results in the Log Service console. This section shows a sample table schema and a sample log that is collected by Logtail.

Table schema

```
CREATE TABLE `VersionOs` (
    'id` int(11) unsigned NOT NULL AUTO_INCREMENT COMMENT 'id',
    'time` datetime NOT NULL,
    'version` varchar(10) NOT NULL DEFAULT '',
    'os' varchar(10) NOT NULL,
    'count` int(11) unsigned NOT NULL,
    PRIMARY KEY (`id`),
    KEY `timeindex` (`time`)
)
```

Sample log

```
"count": "4"
"id: "721097"
"os: "Windows"
"time: "2017-08-25 13:00:00"
"version": "1.3.0"
```

4.3.1.6.3. Collect syslogs

This topic describes how to create a Logtail configuration in the Log Service console to collect syslogs.

Prerequisites

Logtail is installed on the server from which you want to collect syslogs. For more information, see Install Logtail on a Linux server or Install Logtail on a Windows server.

⑦ Note Linux servers support Logtail V0.16.13 or later. Windows servers support Logtail V1.0.0.8 or later.

Overview

Linux servers allow you to use syslog agents such as rsyslog to forward on-premises syslogs to the IP address and port of a specified server. After you apply a Logtail configuration to the specified server, the Logtail plug-in specified in the configuration receives the forwarded syslogs over TCP or UDP. The plug-in also parses the syslogs based on the specified syslog protocol, and extracts the facility, tag(program), severity, and content fields from the syslogs. The syslog protocols defined in RFC 3164 and RFC 5424 are supported. For more information, see RFC 5424 and RFC 3164.

You can configure multiple Logtail plug-ins based on your business requirements. For example, you can configure two Logtail plug-ins to listen on 127.0.0.1:9999 over TCP and UDP.

Implementation

After you configure Logtail plug-ins to listen on a specified address and port, Logtail collects and sends data to Log Service. The data includes the system logs that are collected by using rsyslog, the access logs or error logs that are forwarded by NGINX, and the logs that are forwarded by syslog clients.



Configure Logtail plug-ins to collect syslogs

1. Add a forwarding rule for rsyslog.

- i. Modify the /etc/rsyslog.conf configuration file of rsyslog on the server from which you want to collect syslogs. Add a forwarding rule to the end of the configuration file.
- After the forwarding rule is added, rsyslog forwards syslogs to a specified IP address and port.
- If Logtail resides on the syslog server, you must specify the IP address 127.0.0.1 and a non-well-known port that is unoccupied in the forwarding rule.
- If Logtail resides on a different server from the syslog server, you must specify the public IP address of the different server and a non-well-known
 port that is unoccupied in the forwarding rule.

The following example shows a forwarding rule, which allows all syslogs to be forwarded to 127.0.0.1:9000 over TCP. For more information about the configuration file, see RSyslog Documentation.

. @@127.0.0.1:9000

ii. Run the following command to restart rsyslog and validate the log forwarding rule:

sudo service rsyslog restart

- 2. Log on to the Simple Log Service console
- 3. In the Import Data section, select Custom Data Plug-in.
- 4. Select the project and Logstore. Then, click Next

You can also click Create Now to create a project and a Logstore.

If you click the plus sign (+) next to **Data Import** below your Logstore to enter the configuration process, the system automatically skips this step. 5. Create a machine group and click **Next**.

Before you create a machine group, make sure that Logtail is installed.

Follow the on-screen instructions to install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows.

After you install Logtail, click **Complete Installation** to create a machine group. If you created a machine group, click **Use Existing Machine Groups**.

6. Select the machine group in the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

! Important

If you apply a machine group immediately after you create the machine group, the heartbeat status of the machine group may be **FAIL**. This issue occurs because the machine group is not connected to Simple Log Service. To resolve the issue, you can click **Automatic Retry**. If the status is still **FAIL**, see What do I do if a Logtail machine group has no heartbeats?

ource Server Groups			Applied Server Groups	
Search by server group name	Q		Search by server group name	Q
	d			
		>		
		<		
1 Itama			0 Items	

- 7. In the Specify Data Source step, configure the Config Name and Plug-in Config parameters.
- inputs: specifies the collection configurations. This parameter is required. Configure the inputs parameter based on your data source.

⑦ Note
You can configure only one type of data source in the inputs field.

 processors: specifies the processing method. This parameter is optional. You can configure one or more processing methods in the processors field. For more information, see Customize Logtail plug-ins to process data.

The following example shows how to configure Logtail plug-ins to listen on 127.0.0.1:9000 over UDP and TCP:

Cloud Defined Storage

{		
	"inp	its": [
		"type": "service syslog",
		"detail": {
		"Address": "tcp://127.0.0.1:9000",
		"ParseProtocol": "rfc3164"
)
		"type": "service syslog",
		"detail": {
		"Address": "udp://127.0.0.1:9001",
		"ParseProtocol": "rfc3164"
		}
]	
}		

Parameter	Туре	Required	Description
type	string	Yes	The type of the data source. Set the value to service_syslog .
Address	string	No	 The listening protocol, address, and port that are used by a Logtail plug-in. The plug-in listens on and obtains data based on the Logtail configuration. The value of the parameter is in the [tcp/udp]://[<i>ip</i>]:[<i>port</i>] format. Default value: tcp://127.0.0.1:9999. Note The listening protocol, address, and port that you specify must be the same as those specified in the forwarding rule that is added to the configuration file of rsyslog. If the Logtail server uses multiple IP addresses to receive data, set the IP address to 0.0.0. This address indicates that the plug-in listens on all the IP addresses of the server.
ParseProtocol	string	No	 The protocol that is used to parse syslogs. By default, this parameter is empty. If you leave this parameter empty, the system does not parse syslogs. Valid values: rfc3164: The RFC 3164 protocol is used to parse syslogs. rfc5424: The RFC 5424 protocol is used to parse syslogs. auto: The plug-in automatically selects a protocol based on the content of syslogs.
IgnoreParseFailure	boolean	No	Specifies whether to perform an operation on a syslog after the syslog fails to be parsed. Default value: true . This value true indicates that the system does not parse the syslog and adds the syslog to the content field. If you set the value to false , the syslog is discarded after it fails to be parsed.

8. Configure indexes for query and analysis. After you complete the settings, click Next.

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on the collected logs in manual or automatic mode. For more information, see Enable the index feature and configure indexes for a Logstore.

? Note

- If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.
- If you configure an index for a field of the Long or Double type, you cannot configure Case Sensitive or Delimiter for the field.

Configure Logtail plug-ins to collect NGINX logs

NGINX servers allow you to forward access logs to specified IP addresses and ports by using the syslog protocol. If you want to deliver all data of a server as syslogs to Log Service, you can create a Logtail configuration to collect the data. The data includes NGINX access logs.

1. Add a forwarding rule for NGINX.

i. Add a forwarding rule to the nginx.conf configuration file on the NGINX server. For more information, see NGINX Beginner's Guide.

The following sample code provides an example of a forwarding rule:

```
http {
    ...
    # Add this line.
    access_log syslog:server=127.0.0.1:9000,facility=local7,tag=nginx,severity=info combined;
    ...
}
```

ii. Run the following command to restart the NGINX service and validate the forwarding rule:

sudo service nginx restart

^{2.} Create a Logtail configuration.

For more information, see Configure Logtail plug-ins to collect syslogs.

What to do next

After Logtail collects and sends syslogs to Log Service, you can view the logs in the Log Service console.

1	Q	08-20 16:38:44	_source_:
			_tag :_hostname_:
			topic:
			content: 127.0.0.1 - [20/Aug/2018:16:38:44 +0800] "GET /test.html HTTP/1.1" 404 3650 "-" "curl/7.29.0"
			facility: 23
			hostname:
			ip:
			priority: 190
			program: nginx
			severity: 6
			univtimestamp · 1594754924

Log field	Description
hostname	The hostname. If no hostname is included in the log, the hostname of the current host is obtained.
program	The tag field in the syslog protocol.
priority	The priority field in the syslog protocol.
facility	The facility field in the syslog protocol.
severity	The severity field in the syslog protocol.
unixtimestamp	The timestamp of the log.
content	The content of the log. If the log fails to be parsed, this field contains the content of the raw log.
ip	The IP address of the current host.

4.3.1.6.4. Customize Logtail plug-ins to process data

If you have complex logs that cannot be parsed in basic modes such as full regex, NGINX, and JSON, you can use Logtail plug-ins to parse the logs. You can configure Logtail plug-ins for one or more processing methods. Then, Logtail executes the processing methods in sequence.

Limits

Performance limits

If a plug-in is used to process data, Logtail consumes more resources. Most of these resources are CPU resources. You can modify the Logtail parameter settings based on your business requirements. For more information, see Configure the startup parameters of Logtail.

Limits on text logs

Log Service allows you to process text logs in basic modes such as full regex, NGINX, or JSON. Log Service also allows you to use Logtail plug-ins to process text logs. However, Logtail plug-ins have the following limits on text logs:

- If you enable the plug-in processing feature, some advanced features of the specified mode become unavailable. For example, you cannot
 configure the filter, upload raw logs, specify the system time zone, drop logs that fail to be parsed, or upload incomplete logs in delimiter mode.
- Plug-ins use the line mode to process text logs. In this mode, file-level metadata such as <u>_tag_:_path_</u> and <u>_topic_</u> is stored in each log. If you use Logtail plug-ins to process data, the following limits apply to tag-related features:
- You cannot use the contextual query and LiveTail features because these features depend on fields such as **tag_:_path_**.
- The name of the _topic_ field is renamed to _log_topic_.
- Fields such as **_tag_:_path_** no longer have original field indexes. You must configure indexes for these fields.

Usage notes

When you configure data processing methods, you must set the key in the configuration file to **processors** and set the value to an array of JSON objects. Each object of the array contains the details of a processing method.

Each processing method contains the **type** and **detail** fields. The **type** field specifies the type of the processing method and the **detail** field contains configuration details.

"processors" : [{ "type" : "processor_split_char",
"detail" : {"SourceKey" : "content", "SplitSep": "", "SplitKeys": ["method", "type", "ip", "time", "req_id", "size", "detail"] } }, { "type" : "processor_anchor", "detail" : "SourceKey" : "detail", "Anchors" : [{
 "Start" : "appKey=",
 "_____", "Stop" : ",env=", "FieldName" : "appKey", "FieldType" : "string" }] }]

The following table describes the Logtail plug-ins that are available and the operations that you can perform by using these plug-ins.

Logtail plug-in	Description
processor_regex	You can use the processor_regex plug-in to extract the fields that match a specified regular expression. For more information, see Extract log fields by using a regular expression
processor_anchor	You can use the processor_anchor plug-in to anchor strings and extract fields based on the start and stop keywords that you specify. For more information, see Extract log fields by using start and stop keywords
processor_split_char	You can use the processor_split_char plug-in to extract fields based on a specified single-character delimiter. For more information, see Extract log fields by using a single-character delimiter.
processor_split_string	You can use the processor_split_string plug-in to extract fields based on a specified multi-character delimiter. For more information, see Extract log fields by using a multi-character delimiter.
processor_split_key_value	You can use the processor_split_key_value plug-in to extract fields based on key-value pairs. For more information, see Extract log fields by splitting key-value pairs
processor_add_fields	You can use the processor_add_fields plug-in to add fields to a log. For more information, seeAdd log fields.
processor_drop	You can use the processor_drop plug-in to drop specified fields. For more information, seeDrop log fields.
processor_rename	You can use the processor_rename plug-in to rename specified fields. For more information, seeRename log fields.
processor_packjson	You can use the processor_packjson plug-in to encapsulate one or more fields into a field in the JSON format. For more information, see Encapsulate log fields (JSON).
processor_json	You can use the processor_json plug-in to expand JSON fields. For more information, seeExpand JSON fields.
processor_filter_regex	You can use the processor_filter_regex plug-in to filter logs. For more information, seeFilter logs by using regular expressions.
processor_gotime	You can use the processor_gotime plug-in to extract time information from a field in a time format that is supported by Golangand, and then configure the time information as the log time. For more information, see Extract log time (Go).
processor_strptime	You can use the processor_strptime plug-in to extract time information from a field in a time format that is supported by strptime, and then configure the time information as the log time. For more information, see Extract log time (strptime).
processor_geoip	You can use the processor_geoip plug-in to convert IP addresses in logs to geographical locations. A geographical location includes the following information: country, province, city, longitude, and latitude. For more information, see Convert an IP address to a geographical location

You can also create a custom method that includes one or more of the preceding methods. For more information, see Custom methods.

Extract log fields by using a regular expression

You can use a regular expression to extract log fields.

The type of the plug-in is $\ensuremath{\mathtt{processor_regex}}$.

• Parameters

The following table describes the parameters that you can specify in the **detail** parameter if you set the **type** parameter to **processor_regex**.

Parameter	Туре	Required	Description
SourceKey	String	Yes	The name of the source field.
Regex	String	Yes	The regular expression. Enclose the fields that you want to extract in parentheses () .
Keys	String array	Yes	The array of fields that are extracted, for example, ["ip", "time", "method"].
NoKeyError	Boolean	No	Specifies whether to report an error if a field is not matched. Default value: false. This value indicates that no error is reported if a field is not matched.
NoMatchError	Boolean	No	Specifies whether to report an error if the regular expression does not match the value of a specified field. Default value: false. This value indicates that no error is reported if a field is not matched.
KeepSource	Boolean	No	Specifies whether to retain the source field. Default value: false. This value indicates that the source field is not retained.
FullMatch	Boolean	No	Default value: true. This value indicates that exact match is performed when the regular expression specified in the Regex parameter is used to match field values. If you set the value to false, partial match is performed when the regular expression is used to match field values.

Configuration example

The following example shows how to extract the value of the **content** field. Then, you can set the names of the destination fields to **ip**, **time**, **method**, **url**, **request_time**, **request_length**, **status**, **length**, **ref_url**, and **browser**.

Raw log

```
"content" : "203.0.113.10 - - [10/Aug/2017:14:57:51 +0800] \"POST /PutData?
Category=YunOSAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=
<yourSignature> HTTP/1.1\" 0.024 18204 200 37 \"-\" \"aliyun-sdk-java"
```

• Logtail plug-in configurations for data processing

```
tmethod" : "POST"
"url" : "/PutData?Category=YunOsAccountOpLog&AccessKeyId=
<yourAccessKeyId>&Date=Fri&2C&202&&20Jun&202013&2006&3A53&3A30&20GMT&Topic=raw&Signature=<yourSignature>"
"request_time" : "0.024"
"request_length" : "18204"
"status" : "200"
"length" : "27"
"ref_url" : "-"
"browser" : "aliyun-sdk-java"
```

Extract log fields by using start and stop keywords

You can use start and stop keywords to anchor strings and extract log fields.

The type of the plug-in is processor_anchor .

```
    Parameters
```

The following table describes the parameters that you can specify in the **detail** parameter if you set the **type** parameter to **processor_anchor**.

Parameter	Туре	Required	Description
SourceKey	String	Yes	The name of the source field.
Anchors	Anchor array	Yes	The list of the parameters that are set to anchor strings.

NoAnchorError	Boolean	No	Specifies whether to report an error if no keyword is found. Default value: false. This value indicates that no error is reported if no keyword is found.
NoKeyError	Boolean	No	Specifies whether to report an error if a field is not matched. Default value: false. This value indicates that no error is reported if a field is not matched.
KeepSource	Boolean	No	Specifies whether to retain the source field. Default value: false. This value indicates that the source field is not retained.

The following table describes the parameters of the $\ensuremath{\textbf{Anchors}}$ parameter.

Parameter	Туре	Required	Description
Start	String	Yes	The keyword that anchors the start of a substring in a string. If you do not specify the parameter, the start of the string is matched.
Stop	String	Yes	The keyword that anchors the end of a substring in a string. If you do not specify the parameter, the end of the string is matched.
FieldName	String	Yes	The name of the field that you want to extract.
FieldType	String	Yes	The type of the field that you want to extract. Valid values: string and json .
ExpondJson	Boolean	No	Specifies whether to expand a JSON substring that is anchored. Default value: false. This value indicates that a JSON substring that is anchored is not expanded. This parameter is available only if the value of the FieldType parameter is set to json .
ExpondConnecter	String	No	The character that is used to connect expanded keys. Default value:
MaxExpondDepth	Int	No	The maximum depth of JSON expansion. Default value: 0. This value indicates that the depth of JSON expansion is unlimited.

Configuration example

The following example shows how to extract the value of the **content** field. Then, you can set the names of the destination fields to **time**, **val_key1**, **val_key2**, **val_key3**, **value_key4_inner1**, and **value_key4_inner2**.

• Raw log

"content" : "time:2017.09.12 20:55:36\tjson:{\"key1\" : \"xx\", \"key2\": false, \"key3\":123.456, \"key4\" : { \"inner1\" : 1, \"inner2\" : false}}"

• Logtail plug-in configurations for data processing

```
{
    "type" : "processor_anchor",
    "detail" : {"SourceKey" : "content",
        "Anchors" : [
        {
            "Start" : "time",
            "FieldName" : "time",
            "FieldName" : "string",
            "ExpondJson" : false
        },
        {
            "Start" : "json:",
            "FieldName" : "val",
            "FieldName" : "val",
            "FieldType" : "json",
            "ExpondJson" : true
        }
     ]
     }
        vesult
```

"time": "2017.09.12 20:55:36" "val_key1": "xx" "val_key2": "false" "val_key3": "123.456" "value_key4_inner1": "1" "value_key4_inner2": "false"

Extract log fields by using a single-character delimiter

You can use a specified single-character delimiter to extract fields. This processing method allows you to specify a quote to enclose the delimiter. The type of the plug-in is processor split char.

Parameters

The following table describes the parameters that you can specify in the **detail** parameter if you set the **type** parameter to **processor split char**.

Parameter	Туре	Required	Description
SourceKey	String	Yes	The name of the source field.
SplitSep	String	Yes	The delimiter. The delimiter must be a single character. You can specify a non-printable character as a single-character delimiter, for example, \u0001 .
SplitKeys	String array	Yes	The names of the delimited fields, for example, ["ip","time","method"].
QuoteFlag	Boolean	No	Specifies whether to use a quote to enclose the specified delimiter. Default value: false. This value indicates that a quote is not used to enclose the specified delimiter.
Quote	String	No	The quote. The quote must be a single character. You can specify a non-printable character as a quote, for example, \u0001. This parameter is available only if the value of QuoteFlag is set to true.
NoKeyError	Boolean	No	Specifies whether to report an error if a field is not matched. Default value: false. This value indicates that no error is reported if a field is not matched.
NoMatchError	Boolean	No	Specifies whether to report an error if a delimiter is not matched. Default value: false. This value indicates that no error is reported if a delimiter is not matched.
KeepSource	Boolean	No	Specifies whether to retain the source field. Default value: false. This value indicates that the source field is not retained.

Configuration example

The following example shows how to use a vertical bar () as a delimiter to extract the value of the **content** field. Then, you can set the names of the destination fields to **ip**, **time**, **method**, **url**, **request_time**, **request_length**, **status**, **length**, **ref_url**, and **browser**.

Raw log

```
"content" : "203.0.113.10|10/Aug/2017:14:57:51 +0800|POST|PutData?
Category=Yun0sAccount0pLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=
<yourSignature>|0.024|18204|200|37|-|
aliyun-sdk-java"
```

• Logtail plug-in configurations for data processing

```
{
  "type" : "processor_split_char",
  "detail" : ("SourceKey" : "content",
    "SplitSep" : "|",
    "SplitKeys" : ["ip", "time", "method", "url", "request_time", "request_length", "status", "length", "ref_url", "browser"]
  }
}
```

Result

```
"ip" : "203.0.113.10"
"time" : "10/Aug/2017:14:57:51 +0800"
"method" : "POST"
"url" : "/PutData?Category=Yun0sAccountOpLog&AccessKeyId=
<yourAccessKeyId>KDate=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=<yourSignature>"
"request_time" : "0.024"
"request_length" : "18204"
"status" : "200"
"length" : "27"
"horoser" : "aliyun-sdk-java"
```

Extract log fields by using a multi-character delimiter

You can use a specified multi-character delimiter to extract fields. You cannot specify a quote to enclose the delimiter.

The type of the plug-in is processor_split_string .

Parameters

The following table describes the parameters that you can specify in the **detail** parameter if you set the **type** parameter to **processor_split_string**.

Parameter	Туре	Required	Description
SourceKey	String	Yes	The name of the source field.
SplitSep	String	Yes	The delimiter. The delimiter contains multiple characters. You can specify non-printable characters in the delimiter, for example, \u0001\u0002.
SplitKeys	String array	Yes	The names of the delimited fields, for example, ["key1", "key2"].
PreserveOthers	Boolean	No	Specifies whether to retain excess fields if the number of fields is greater than the number of fields that are specified by the SplitKeys parameter. Default value: false. This value indicates that excess fields are not retained.
ExpandOthers	Boolean	No	Specifies whether to parse excess fields. Default value: false. This value indicates that excess fields are not parsed.
ExpandKeyPrefix	String	No	The name prefix of excess fields. For example, if you specify expand_ for the parameter, the first two excess fields are named expand_1 and expand_2 .
NoKeyError	Boolean	No	Specifies whether to report an error if a field is not matched. Default value: false. This value indicates that no error is reported if a field is not matched.
NoMatchError	Boolean	No	Specifies whether to report an error if a delimiter is not matched. Default value: false. This value indicates that no error is reported if a delimiter is not matched.
KeepSource	Boolean	No	Specifies whether to retain the source field. This value indicates that the source field is not retained.

• Configuration example

The following example shows how to use a delimiter (|#|) to extract the value of the **content** field. Then, you can set the names of the destination fields to **ip**, **time**, **method**, **url**, **request_time**, **request_length**, **status**, **expand_1**, **expand_2**, and **expand_3**.

```
    Raw log
```

```
"content" : "203.0.113.10|#|10/Aug/2017:14:57:51 +0800|#|POST|#|PutData?
Category=YunOSAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=
<yourSignature>|#|0.024|#|18204|#|200|#|27|#|-|#|
aliyun-sdk-java"
```

• Logtail plug-in configurations for data processing

```
{
  "type": "processor_split_string",
  "detail": {"SourceKey": "content",
    "SplitSep": "!#!",
    "SplitKeys": ["ip", "time", "method", "url", "request_time", "request_length", "status"],
    "PreserveOthers": true,
    "ExpandOthers": true,
    "ExpandKeyPrefix": "expand_"
}
```

• Result

```
"ip" : "203.0.113.10"
"time" : "10/Aug/2017:14:57:51 +0800"
"method" : "POST"
"url" : "/PutData?Category=YunOsAccountOpLog&AccessKeyId=
<yourAccessKeyId>&Shate=Fri&2C&2028&20Jun&202013&2006&3A53&3A30&20GMT&Topic=raw&Signature=<yourSignature>"
"request_time" : "0.024"
"request_length" : "18204"
"status" : "200"
"expand_1" : "27"
"expand_2" : "-"
"expand_3" : "aliyun-sdk-java"
```

Extract log fields by splitting key-value pairs

You can split key-value pairs to extract log fields.

The type of the plug-in is <code>processor_split_char</code> .

Parameters

The following table describes the parameters that you can specify in the **detail** parameter if you set the **type** parameter to **processor_split_key_value**.

? Note

Only Logtail V0.16.26 or later supports the plug-in.

Parameter	Туре	Required	Description
SourceKey	String	Yes	The name of the source field.
Delimiter	String	No	The delimiter between key-value pairs. Default value: \t .
Separator	String	No	The delimiter that is used to separate the key and the value in a single key-value pair. A colon (:) is used by default.
KeepSource	Boolean	No	Specifies whether to retain the source field. Default value: true. This value indicates that the source field is retained.
ErrifSourceKeyNotFound	Boolean	No	Specifies whether to trigger an alert if a field is not matched. Default value: true. This value indicates that an alert is triggered if a field is not matched.
DiscardWhenSeparatorNot Found	Boolean	No	Specifies whether to drop the key-value pair if a field is not matched. Default value: false. This value indicates that the key-value pair is not dropped if a field is not matched.
ErrifSeparatorNotFound	Boolean	No	Specifies whether to trigger an alert if the delimiter specified by the Separator parameter does not exist. Default value: true. This value indicates that an alert is triggered if the specified delimiter does not exist.

Configuration example

The following example shows how to split the key-value pairs in the value of the **content** field. The delimiter that is used to separate key-value pairs is a tab character (/t). The delimiter that is used to separate the key and the value in a single key-value pair is a colon (:).

Raw log

"content": "class:main\tuserid:123456\tmethod:get\tmessage:\"wrong user\""

Logtail plug-in configurations for data processing

```
{
    "processors":[
    {
        "type":"processor_split_key_value",
        "detail": {
            "SourceKey": "content",
            "Delimiter": "\t",
            "Separator": ":",
            "KeepSource": true
        }
    }
}
```

Result

"content": "class:main\tuserid:123456\tmethod:get\tmessage:\"wrong user\""
"class": "main"
"userid": "123456"
"method": "get"
"message": "\"wrong user\""

Convert an IP address to a geographical location

This processing method converts IP addresses in logs to geographical locations. A geographical location includes the following information: country, province, city, longitude, and latitude.

The type of the plug-in is <code>processor_geoip</code> .

? Note

- GeoIP databases are not included in the Logtail installation package. You must download and configure a GeoIP database on the server where Logtail is installed. We recommend that you download a database that provides the **city** information of an IP address.
- Make sure that the database format is MMDB.

Parameters

The following table describes the parameters that you can specify in the detail parameter if you set the type parameter to processor_geoip.

Parameter	Туре	Required	Description
SourceKey	String	Yes	The name of the source field that you want to convert.

DBPath	String	Yes	The absolute path of the GeoIP database, for example, /user/data/GeoLite2-City_20180102/GeoLite2-City.mmdb.
NoKeyError	Boolean	No	Specifies whether to report an error if a field is not matched. Default value: false. This value indicates that no error is reported if a field is not matched.
NoMatchError	Boolean	No	Specifies whether to report an error if an IP address is invalid or is not matched in the database. Default value: false. This value indicates that no error is reported if an IP address is invalid or is not matched in the database.
KeepSource	Boolean	No	Specifies whether to retain the source field. Default value: true. This value indicates that the source field is retained.
Language	String	No	The language of the GeoIP database. Default value: zh-CN . Make sure that your GeoIP database can be displayed in a language that is suitable for your business.

Configuration example

The following example shows how to configure the processing method to convert IP addresses in logs to geographical locations.

Raw log

```
"source_ip" : "203.0.113.10"
```

• Logtail plug-in configurations for data processing

```
{
  "type": "processor_geoip",
  "detail": {
    "SourceKey": "ip",
    "NoKeyError": true,
    "NoMatchError": true,
    "KeepSource": true,
    "DBPath" : "/user/local/data/GeoLite2-City_20180102/GeoLite2-City.mmdb"
    }
}
```

• Result

"source_ip_city_" : "**.**.**"
"source_ip_province_" : "Zhejiang"
"source_ip_city_" : "Hangzhou"
"source_ip_contry_code_" : "ZJ"
"source_ip_contry_code_" : "CN"
"source_ip_longitude_" : "120.********"

Filter logs by using regular expressions

 $\label{eq:result} This method uses regular expressions to filter logs. You can specify conditions in the $$Include$ and $$Exclude$ parameters.$

The type of the plug-in is processor_filter_regex .

• Parameters

The following table describes the parameters that you can specify in the **detail** parameter if you set the **type** parameter to **processor_filter_regex**.

⑦ Note

A log is collected only if the log exactly matches the regular expression that is specified in the **Include** parameter and does not match the regular expression that is specified in the **Exclude** parameter.

Parameter	Туре	Required	Parameters
Include	JSON object that conatins key- value pairs	No	A map that includes key-value pairs. In each key-value pair, the key specifies a field and the value specifies a regular expression that the value of the same field in each log must match. If the values of all fields in a log match the regular expressions that are specified in the Include parameter, the log is collected.
Exclude	JSON object that conatins key- value pairs	No	A map that includes key-value pairs. In each key-value pair, the key specifies a field and the value specifies a regular expression that the value of the same field in each log must match. If the values of all fields in a log match the regular expressions that are specified in the Exclude parameter, the log is not collected.

Configuration example

The following example shows how to use regular expressions to filter logs.

- Raw logs
- Log 1

"ip" : "203.0.113.10" "method" : "POST"

"browser" : "aliyun-sdk-java"

Log 2

"ip" : "203.0.113.20" "method" : "POST" ... "browser" : "chrome"

Log 3

"ip" : "198.51.100.10" "method" : "POST" ... "browser" : "ali-sls-ilogtail"

• Logtail plug-in configurations for data processing

```
{
    "type" : "processor_filter_regex",
    "detail" : {
        "Include" : {
            "ip" : "203\\..*",
            "method" : "POST"
        },
        "Exclude" : {
            "browser" : "aliyun.*"
        }
    }
}
```

• Result

Log	Collected	Reason
Log 1	No	The value of the browser parameter matches the regular expression that is specified in the Exclude parameter.
Log 2	Yes	All the filter conditions are met.
Log 3	No	The value of the ip parameter does not match the regular expression that is specified in the Include parameter.

Add log fields

You can use this method to add multiple fields to a log.

The type of the plug-in is processor_add_fields .

• Parameters

The following table describes the parameters that you can specify in the **detail** parameter if you set the **type** parameter to **processor_add_fields**.

 Note Only Logtail V0.16.28 or later supports the plug-in. 				
Parameter	Туре	Required	Description	
Fields	Мар	No	The key-value pairs that you want to add. You can specify multiple key-value pairs in the parameter.	
IgnorelfExist	Boolean	No	Specifies whether to retain key-value pairs that have the same key. Default value: false. This value indicates that a key-value pair is not retained if the key is the same as another specified key.	

Configuration example

The following example shows how to add the **aaa2** and **aaa3** fields to a log.

Raw log

"aaa1":"value1"

• Logtail plug-in configurations for data processing

```
{
    "processors":[
        {
          "type":"processor_add_fields",
          "detail": {
              "Fields": {
                  "aaa2": "value2",
                   "aaa3": "value3"
        }
        }
    }
    Result
    "aaa1":"value1"
```

"aaa2":"value2" "aaa3":"value3"

Drop log fields

You can use this method to drop specified fields from a log.

The type of the plug-in is $processor_drop$.

Parameters

The following table describes the parameters that you can specify in the **detail** parameter if you set the **type** parameter to **processor_drop**.

⑦ Note Only Logtail V0.16.28 or later supports the plug-in.			
Parameter	Туре	Required	Description
DropKeys	String array	No	The fields that you want to drop. You can drop one or more fields from a log.

Configuration example

The following example shows how to drop the **aaa1** and **aaa2** fields from a log.

Raw log

```
"aaa1":"value1"
"aaa2":"value2"
"aaa3":"value3"
```

• Logtail plug-in configurations for data processing

• Result

"aaa3":"value3"

Extract log time (Go)

You can use this method to extract time information from a specified field, and then convert the time format.

The type of the plug-in is processor_gotime .

Parameters

The following table describes the parameters that you can specify in the **detail** parameter if you set the **type** parameter to **processor_gotime**.

⑦ Note Only Logtail V0.16.28 or later supports the plug-in.				
Parameter	Туре	Required	Description	
SourceKey	String	Yes	The name of the source field.	
SourceFormat	String	Yes	The format of the time information in the source field.	

SourceLocation	Int	Yes	The source time zone. If you do not specify the parameter, the current time zone of the server where Logtail is installed is used.
DestKey	String	Yes	The name of the destination field.
DestFormat	String	Yes	The format of the time information in the destination field.
DestLocation	Int	No	The destination time zone. If you do not specify the parameter, the current time zone of the server where Logtail is installed is used.
SetTime	Boolean	No	Specifies whether to configure the time information as the log time. Default value: true. This value indicates that the time information is configured as the log time.
KeepSource	Boolean	No	Specifies whether to retain the source field. Default value: true. This value indicates that the source field is retained.
NoKeyError	Boolean	No	Specifies whether to report an error if a field is not matched. Default value: true. This value indicates that an error is reported if the source field is not matched.
AlarmlfFail	Boolean	No	Specifies whether to trigger an alert if the time information fails to be extracted. Default value: true. This value indicates that an alert is triggered if the time information fails to be extracted.

Configuration example

In this example, the time information 2006-01-02 15:04:05 (UTC+8) is extracted from the **s_key** field, converted to 2006/01/02 15:04:05 (UTC+9) , and then added to the **d_key** field.

Raw log

"s_key":"2019-07-05 19:28:01"

• Logtail plug-in configurations for data processing



• Result

"s_key":"2019-07-05 19:28:01" "d_key":"2019/07/05 20:28:01"

Expand JSON fields

You can use this method to expand a JSON field.

The type of the plug-in is $processor_json$.

Parameters

÷.

The following table describes the parameters that you can specify in the **detail** parameter if you set the **type** parameter to **processor_json**.

NoteOnly Logtail V0.16.28 or later supports the plug-in.				
Parameter	Туре	Required	Description	
SourceKey	String	Yes	The name of the source field.	

NoKeyError	Boolean	No	Specifies whether to report an error if the source field is not matched. Default value: true. This value indicates that an error is reported if the source field is not matched.
ExpandDepth	Int	No	The depth of JSON expansion. Default value: 0. This value indicates that the depth of JSON expansion is unlimited. If the value is n, the depth of JSON expansion is n.
ExpandConnector	String	No	The character that is used to connect expanded keys. You can leave this parameter empty. Default value:
Prefix	String	No	The prefix that is added to expanded keys. You can leave this parameter empty.
KeepSource	Boolean	No	Specifies whether to retain the source field. Default value: true. This value indicates that the source field is retained.
UseSourceKeyAsPrefix	Boolean	No	Specifies whether to add the name of the source field as a prefix to all expanded keys. Default value: false. This value indicates that the name of the source field is not added.

Configuration example

The following example shows how to expand the JSON field **s_key**, and then add **j** and the name of the source field **s_key** as a prefix to the expanded keys.

Raw log

"s_key":"{\"k1\":{\"k2\":{\"k3\":{\"k4\":{\"k51\":\"51\",\"k52\":\"52\"},\"k41\":\"41\"}}})"

• Logtail plug-in configurations for data processing

```
"processors":[
    {
        "type":"processor_json",
        "detail": {
            "SourceKey": "s_key",
            "NKeyError":true,
            "ExpandDepth":0,
            "ExpandDepth":0,
            "ExpandConnector":"-",
            "Prefix":"j",
            "Prefix":"j",
            "KeepSource": false,
            "UseSourceKeyAsPrefix": true
        }
    }
]
```

Result

```
"s_key":"{\"k1\":{\"k2\":{\"k4\":{\"k51\":\"51\",\"k52\":\"52\"},\"k41\":\"41\"}})"
"js_key-k1-k2-k3-k4-k51":"51"
"js_key-k1-k2-k3-k4-k52":"52"
"js_key-k1-k2-k3-k41":"41"
```

Encapsulate log fields (JSON)

You can use this method to encapsulate one or more fields into a field in the JSON format.

The type of the plug-in is $\ensuremath{\mathtt{processor_packjson}}$.

• Parameters

The following table describes the parameters that you can specify in the **detail** parameter if you set the **type** parameter to **processor_packjson**.

Note Only Logtail V0.16.28 or later supports the plug-in.				
Parameter	Туре	Required	Description	
SourceKeys	String array	Yes	The field that you want to encapsulate. The field is in the string array format.	
DestKey	String	No	The destination field in the JSON format.	
KeepSource	Boolean	No	Specifies whether to retain the source field. Default value: true. This value indicates that the source field is retained.	

User Guide-Log Service

	AlarmIfincomplete	Boolean	No	Specifies whether to trigger an alert if the source field does not exist. Default value: true. This value indicates that an alert is triggered if the source field does not exist.		
• (Configuration example					
٦	he following example shows	how to encapsulate the a and	b fields into the d_key field.			
0	Raw log					
	"a":"1" "b":"2"					
0	Logtail plug-in configuration	is for data processing				
	<pre>{ "processors":[{ "type":"processor_packjson", "detail": { "SourceKeys": ["a", "b"], "DestKey":"d_key", "BestKey":"d_key", "KeepSource":true, "AlarmIfEmpty":true } } }</pre>					
0	Result					
	"a":"1" "b":"2" "d_key":"{\"a\":\"1\", \"b\":\"2\"}"					
Re	Rename log fields					

You can use this method to rename multiple fields.

The type of the plug-in is <code>processor_rename</code> .

• Parameters

.

The following table describes the parameters that you can specify in the **detail** parameter if you set the **type** parameter to **processor_rename**.

	?	Note	
--	---	------	--

Only Logtail V0.16.28 or later supports the plug-in.

Parameter	Туре	Required	Description
NoKeyError	Boolean	Yes	Specifies whether to report an error if a field that you want to rename is not matched. Default value: false. This value indicates that no error is reported if a field that you want to rename is not matched.
SourceKeys	String array	Yes	The source fields that you want to rename.
DestKeys	String array	Yes	The fields that are renamed.

Configuration example

The following example shows how to rename the **aaa1** field to **bbb1** and the **aaa2** field to **bbb2**.

Raw log

"aaa1":"value1" "aaa2":"value2" "aaa3":"value3"

• Logtail plug-in configurations for data processing

```
{
   "processors":[
   {
        "type":"processor_rename",
        "detail": {
            "SourceKeys": ["aaa1","aaa2"],
            "DestKeys": ["bbb1","bbb2"],
            "NoKeyError": true
        }
    }
   ]
}
```

Cloud Defined Storage

• Result

```
"bbb1":"value1"
"bbb2":"value2"
"aaa3":"value3"
```

Extract log time (strptime)

You can use this method to extract time information from a field, and then configure the time information as the log time.

The type of the plug-in is $\ensuremath{\mbox{processor_strptime}}$.

Parameters

The following table describes the parameters that you can specify in the **detail** parameter if you set the **type** parameter to **processor_strptime**.

Note Only Logical V0 16 28 or later supports the planet.

Only	Logtail	V0.16.28	or	later	supports	the	plug-in.	

Parameter	Туре	Required	Description
SourceKey	String	Yes	The name of the source field.
Format	String	Yes	The format of the time information in the source field.
AdjustUTCOffset	Boolean	No	Specifies whether to modify the time zone. Default value: false. This value indicates that the time zone is not modified.
UTCOffset	Int	No	The offset that is used to modify the time zone. For example, the value 28800 indicates that the time zone is modified to UTC+8.
AlarmifFail	Boolean	No	Specifies whether to trigger an alert if the time information fails to be extracted. Default value: true. This value indicates that an alert is triggered if the time information fails to be extracted.
KeepSource	Boolean	No	Specifies whether to retain the source field. Default value: true. This value indicates that the source field is retained.

• Configuration examples

The following examples show how to parse the value of the **log_time** field into the **vy**/%m/%d %H:%M:%s format. The current time zone of the server where Logtail is installed is used.

- Example 1: The time zone is UTC+8.
 - Raw log

"log_time":"2016/01/02 12:59:59"

Logtail plug-in configurations for data processing

```
{
  "processors":[
   {
      "type":"processor_strptime",
      "detail": {
           "SourceKey": "log_time",
           "Format": "%Y/%m/%d %H:%M:%S"
      }
   }
}
```

Result

```
"log_time":"2016/01/02 12:59:59"
Log.Time = 1451710799
```

- $\circ~$ Example 2: The time zone is UTC+7.
- Raw log

"log_time":"2016/01/02 12:59:59"

Logtail plug-in configurations for data processing

```
{
   "processors":[
    {
        "type":"processor_strptime",
        "detail": {
            "SourceKey": "log_time",
            "Format": "%Y/%m/%d %H:%M:%S",
            "AdjustUTCOffset": true,
            "UTCOffset": 25200
        }
    }
    }
}
```

Result

```
"log_time":"2016/01/02 12:59:59"
Log.Time = 1451714399
```

Custom methods

You can use multiple processing methods to process logs. The following example shows how to use a single-character delimiter to split a log into several fields and then specify anchor points to extract content from the detail field.

```
    Raw log
```

```
"content" :
"ACCESS[QAS]203.0.113.10|1508729889935|52460dbed4d540b88a973cf5452b1447|1238|appKey=ba,env=pub,requestTime=1508729889913,latency=22ms,
request=(appKey:ba,optional:{\\domains\\:\\domains\\:\\version\\:\\v2\\},rawQuery:{\\The route to Location
A\\,\\domain\\:\\Navigation\\,\\intent\\:\\navigate\\,\\slots\\:\\to_geo:level3=Location A\\,\\location\\:\\Location B\\},
requestId:52460dbed4d540b88a973cf5452b1447},
response={answers:[],status:SUCCESS}!"
```

Logtail plug-in configurations for data processing

```
"processors" : [
           {
                "type" : "processor_split_char",
                "detail" : {"SourceKey" : "content",
                    "SplitSep" : "|",
                    "SplitKeys" : ["method", "type", "ip", "time", "req_id", "size", "detail"]
               }
           },
           {
                "type" : "processor_anchor",
                "detail" : "SourceKey" : "detail",
                    "Anchors" : [
                         {
                                   "Start" : "appKey=",
                              "Stop" : ",env=",
                              "FieldName" : "appKey",
"FieldType" : "string"
                          },
                          {
                               "Start" : ",env",
                              "Stop" : ",requestTime=",
"FieldName" : "env",
"FieldType" : "string"
                          },
                          {
                               "Start" : ",requestTime=",
"Stop" : ",latency",
"FieldName" : "requestTime",
                               "FieldType" : "string"
                          },
                          {
                               "Start" : ",latency=",
"Stop" : ",request=",
"FieldName" : "latency",
                               "FieldType" : "string"
                          },
                              "Start" : ",request=",
"Stop" : ",response=",
                               "FieldName" : "request",
                               "FieldType" : "string"
                          },
                          {
                               "Start" : ",response=",
                               "Stop" : "",
                               "FieldName" : "response",
                               "FieldType" : "json"
                          }
                   ]
               }
         }
      ]

    Result

    "method" : "ACCESS"
    "type" : "QAS"
"ip" : "203.0.113.10"
"time" : "1508729889935"
    "req_id" : "52460dbed4d540b88a973cf5452b1447"
    "size" : "1238"
   "appKey" : "ba"
"env" : "pub"
```

```
"size": "1238"
"appKey": "ba"
"env": "pub"
"requestTime": "1508729889913"
"latency": "22ms"
"request": "{appKey:nui-banma,optional:{\\domains\\:\\daily=faq\\,\\version\\:\\v2\\},rawQuery:
{\\query\:\\\345\216\273\344\271\220\345\261\261\347\232\204\350\267\257\347\272\277\\,\\domain\\:\\\345\257\274\350\210\252\\,\\intent\\:\\n
te\,\\slots\\:\\to_geo:level3=\344\271\220\345\261\261\,\\location\\:\\\345\214\227\344\272\254\\},requestId:52460dbed4d540b88a973cf5452b144'
"response_answers": "[]"
```

```
"response_status" : "SUCCESS"
```

4.3.1.7. Limits

This topic describes the limits of Logtail. These limits apply when you collect files, manage resources, and resolve errors.

Limits on file collection

Item	Description
File encoding	Log files can be encoded in UTF-8 and GBK. To improve processing performance, we recommend that you encode log files in UTF-8. If log files are encoded in other formats, errors such as garbled characters and data loss may occur.

Log file size	Unlimited.
Log file rotation	Supported. Both \log^* and \log are supported for file names.
Log collection behavior when log parsing is blocked	When log parsing is blocked, Logtail keeps the log file descriptor (FD) open. If log file rotation occurs multiple times during the blocking period, Logtail attempts to parse new log files in sequence. If the number of new log files that are not parsed exceeds 20, Logtail does not process the exceeds log files.
Symbolic link	Monitored directories can be symbolic links.
Size of a single log	The maximum size of a single log is 512 KB. If a regular expression is used to split a multi-line log to match the start part in the first line of the log, the maximum size of each log after splitting is still 512 KB. If the size of a log exceeds 512 KB, the log is forcibly split into multiple parts and collected. For example, if the size of a log is 1,025 KB, the log is split into three parts of the following sizes: 512 KB, 512 KB, and 1 KB. Then, the log parts are collected in sequence.
Regular expression	Perl-based regular expressions can be used.
Multiple Logtail configurations for the same log file	Not supported. We recommend that you collect and store log files to one Logstore, and then configure multiple subscriptions. If this feature is required, configure symbolic links for log files to bypass this limit.
File opening behavior	When Logtail collects data from a log file, Logtail keeps the log file open. If the log file is not updated for more than 5 minutes and log rotation does not occur, Logtail closes the log file.
First log collection behavior	Logtail collects data only from incremental log files. If the size of a log file exceeds 1 MB the first time an update to the log file is detected, Logtail collects data from the last 1 MB. If the log file size does not exceed 1 MB, Logtail collects data from the beginning of the log file. If the log file is not updated after the Logtail configuration is delivered, Logtail does not collect data from the log file.
Non-standard text logs	If a log contains $\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$

Limits on checkpoints

Item	Description
Checkpoint timeout period	If a log file is not updated for more than 30 days, the checkpoint of the log file is deleted.
Checkpoint storage policy	Checkpoints are saved every 15 minutes and are automatically saved when you exit Logtail.
Checkpoint storage path	By default, checkpoints are stored in the /tmp/logtail_checkpoint directory. You can modify the values of the related parameters. For more information, see Configure the startup parameters of Logtail

Limits on configurations

Item	Description
Configuration update	A custom configuration update requires approximately 30 seconds to take effect.
Dynamic loading of Logtail configurations	Supported. The update of a Logtail configuration does not affect other Logtail configurations.
Number of Logtail configurations	Unlimited. However, we recommend that you create a maximum of 100 Logtail configurations on a server.
Multi-tenant isolation	Logtail configurations for different tenants are isolated.

Limits on resources and performance metrics

Item	Description
Throughput for log processing	The default transmission speed of raw logs is limited to 2 MB/s. Log data is uploaded after it is encoded and compressed. The compression ratio ranges from 5:1 to 10:1. If the transmission speed exceeds the limit, log data may be lost. You can modify the values of the related parameters. For more information, see Configure the startup parameters of Logtail.
Maximum processing speed	Single-core processing speed: The maximum processing speed is 100 MB/s for logs in simple mode, 40 MB/s for logs in delimiter mode, and 30 MB/s for logs in JSON mode. By default, the maximum processing speed is 20 MB/s for logs in full regex mode based on the complexity of regular expressions. If multiple processing threads are enabled, the performance can be improved by 1.5 to 3 times.

Number of monitored directories	Logtail limits the depth of monitored directories to reduce the consumption of your resources. If the upper limit is reached, Logtail stops monitoring additional directories or log files. Logtail can monitor a maximum of 3,000 directories, including subdirectories.
Number of monitored files	 By default, you can use a Logtail configuration on each server to monitor a maximum of 10,000 files. By default, a Logtail client on each server can monitor a maximum of 100,000 files. Excessive files are not monitored. If the upper limit is reached, you can perform the following operations: Improve the depth of the monitored directory in each Logtail configuration. Increase the value of the mem_usage_limit parameter to raise the threshold of memory resources that are available for Logtail. For more information, see Configure the startup parameters of Logtail You can raise the threshold to a maximum of 2 GB. This way, the maximum number of files that can be monitored by using each Logtail configuration is increased to 100,000, and the maximum number of files that the Logtail client on each server can monitor is increased to 1,000,000.
Default resources	By default, Logtail occupies a maximum of 40% of the CPU and 256 MB of memory. If logs are generated at a high speed, you can modify the values of the related parameters. For more information, see Configure the startup parameters of Logtail.
Processing policy of threshold-crossing resources	If the resources that are occupied by Logtail exceed the upper limit and this issue lasts for 5 minutes or more, Logtail is forcibly restarted. The restart may cause data loss or duplication.

Limits on error handling

Item	Description
Network error handling	If a network error occurs, Logtail automatically retries and adjusts the retry interval.
Processing policy of threshold-crossing resources	If the data transmission speed exceeds the quota of the Logstore, Logtail restricts the log collection speed and retries the log collection.
Maximum retry period before timeout	If data fails to be transmitted and the issue lasts for more than six consecutive hours, Logtail discards the data.
Status self-check	Logtail restarts if an exception occurs, for example, an application unexpectedly exits or the resource usage exceeds the quota.

Other limits

Item	Description
Log collection latency	A latency of less than 1 second exists between the point in time when a log is written to a disk and the point in time when Logtail collects the log. However, if the log collection speed is restricted, the latency increases.
Log upload policy	Before Logtail uploads logs, Logtail aggregates the logs in the same file. The log upload starts if the number of logs exceeds 2,000, the total size of logs exceeds 2 MB, or the log collection duration exceeds 3 seconds.

4.3.2. Other collection methods

4.3.2.1. Use the web tracking feature to collect logs

Log Service provides the web tracking feature that you can use to collect logs from the HTML, HTML5, iOS, and Android platforms. You can also customize dimensions and metrics to collect logs. This topic describes how to use the web tracking feature to collect logs.

Background information

- You can use the web tracking feature to collect user information from browsers, iOS apps, or Android apps. The information includes:
- Browsers, operating systems, and resolutions that are used by users.
- User browsing behavior, such as the number of clicks and purchases on a website.
- The amount of time that users spend on an app and whether users are active users.

Usage notes

- After you enable the web tracking feature for a Logstore, the write permissions on the Logstore are granted to anonymous users from the Internet. This may generate dirty data.
- The HTTP body of each GET request cannot exceed 16 KB.
- You can use the POST method to call the PutLogs API operation and write a maximum of 3 MB or 4,096 log entries to Log Service.

Step 1: Enable the web tracking feature

You can use the Log Service console or an SDK to enable the web tracking feature.

- Enable the web tracking feature in the Log Service console.
- i. Log on to the Log Service console.
- ii. In the **Projects** section, click the project in which you want to enable the web tracking feature for a Logstore.
- iii. Find the Logstore for which you want to enable the web tracking feature and choose $_{RR}$ > Modify.

iv. In the upper-right corner of the Logstore Attributes page, click Modify. v. Turn on WebTracking and click Save. • Use an SDK to enable the web tracking feature. The following script shows how to use Log Service SDK for Java to enable the web tracking feature: import com.aliyun.openservices.log.Client; import com.aliyun.openservices.log.common.LogStore; import com.aliyun.openservices.log.exception.LogException; public class WebTracking { static private String accessId = "your accesskey id"; static private String accessKey = "your accesskey"; static private String project = "your project"; static private String host = "log service data address"; static private String logStore = "your logstore"; static private Client client = new Client(host, accessId, accessKey); public static void main(String[] args) { try { // Enable the web tracking feature for an existing Logstore. LogStore logSt = client.GetLogStore(project, logStore).GetLogStore(); client.UpdateLogStore(project, new LogStore(logStore, logSt.GetTtl(), logSt.GetShardCount(), true)); // Disable the web tracking feature. //client.UpdateLogStore(project, new LogStore(logStore, logSt.GetTtl(), logSt.GetShardCount(), false)); $\ensuremath{{\prime}}\xspace$ // Create a Logstore for which you want to enable the web tracking feature. //client.UpdateLogStore(project, new LogStore(logStore, 1, 1, true)); } catch (LogException e) { e.printStackTrace(); } } }

Step 2: Collect logs

After you enable the web tracking feature for a Logstore, you can upload logs to a Logstore by using the following methods:

• Use SDK for JavaScript to upload logs.

i. Install the dependency.

npm install --save js-sls-logger

ii. Import the application module.

import SlsWebLogger from 'js-sls-logger'

iii. Set the **opts** parameter. The following table describes the parameters.

```
const opts = {
    host: 'cn-qingdao-env12-d01.sls-pub.cloud.env12.shuguang.com',
    project: 'my_project_name',
    logstore: 'my_logstore_name',
    time: 10,
    count: 10,
}
```

Parameter	Required	Description
host	Yes	The endpoint of the region where Log Service resides. In this example, the endpoint of the China (Hangzhou) region is used. Replace the value of the parameter with the actual endpoint. For more information, see the Obtain an endpoint topic in Log Service Developer Guide .
project	Yes	The name of the project.
logstore	Yes	The name of the Logstore.
time	No	The time interval at which logs are sent. Default value: 10. Unit: seconds.
count	No	The number of logs that are sent. Default value: 10.

iv. Create SIsWebLogger.

const logger = new SlsWebLogger(opts)

v. Upload logs.

```
logger.send({
    customer: 'zhangsan',
    product: 'iphone 12',
    price: 7998
})
```

• Use the GET method to upload logs.

Run the following command to upload logs. Replace the values of the parameters based on your business requirements. The following table describes the parameters.

curl --request GET 'http://\${project}.\${host}/logstores/\${logstore}/track?APIVersion=0.6.0&key1=val1&key2=val2'

Parameter	Required	Description
\${project}	Yes	The name of the project.
\${host}	Yes	The endpoint of the region where Log Service resides. For more information, see the Obtain an endpoint topic in Log Service Developer Guide .
\${logstore}	Yes	The name of the Logstore.
APIVersion=0.6.0	Yes	A reserved parameter.
topic=yourtopic	No	The topic of the log that you want to upload.
key1=val1&key2=val2	Yes	The key-value pairs that you want to upload to Log Service. Make sure that the data size is less than 16 KB.

• Use HTML tags to upload logs.

The track_ua.gif file contains custom parameters that you want to upload to Log Service. If you use this method to upload logs, Log Service records the custom parameters and the User-Agent and Referer HTTP headers as log fields.

```
? Note
```

To collect the Referer HTTPS header, make sure that the URL in the preceding tag uses the HTTPS protocol.

```
    Use the POST method to upload logs
```

You can send an HTTP POST request to upload a large amount of data. For more information, see the "PutWebtacking" topic of **API Reference** in **Log Service Developer Guide**.

4.3.2.2. Use SDKs to collect logs

4.3.2.2.1. Producer Library

The Aliyun LOG Java Producer supports Java applications that run in big data processing scenarios with high concurrency. The library is easy to use and highly customizable.

For more information about the related GitHub project, visit Aliyun LOG Java Producer.

4.3.2.2.2. Log4j Appender

This topic describes Alibaba Cloud Log4j Appender.

Log4j is an open source project of Apache. Log4j allows you to specify the output destination and format of logs. You can also specify the severity level of each log for fine-grained control on log generation. Log4j consists of the following three components:

- Loggers
- The severity levels of logs are classified into ERROR, WARN, INFO, and DEBUG in descending order.
- Appenders
 - An appender specifies that logs are sent to the Log Service console or files.
- Layouts
- A layout specifies the output format of logs.

You can use Alibaba Cloud Log4j Appender to send logs to Log Service. For more information about Alibaba Cloud Log4j Appender, visit Log4j Appender.

4.3.2.2.3. Logback Appender

This topic describes how to write logs to Log Service by using Aliyun Log Logback Appender.

Logback is an open source project that is developed by the founder of Log4j. Logback allows you to write logs to multiple destinations. These destinations include the Log Service console, files, graphical user interface (GUI) components, socket servers, NT kernel loggers, and UNIX syslog daemons. You can specify the output format of each log. You can also specify the severity level of each log for fine-grained control on log generation. The following example shows the format of a log that is written to Log Service by using Aliyun Log Logback Appender:

```
level: ERROR
location: com.aliyun.openservices.log.logback.example.LogbackAppenderExample.main(LogbackAppenderExample.java:18)
message: error log
throwable: java.lang.RuntimeException: xxx
thread: main
time: 2018-01-02T03:15+0000
log: 2018-01-02 11:15:29,682 ERROR [main] com.aliyun.openservices.log.logback.example.LogbackAppenderExample: error log
__source_: xxx
__topic_: yyy
```

For more information about Aliyun Log Logback Appender, see Logback Appender.

4.3.2.2.4. Golang Producer Library

The Aliyun LOG Go Producer Library supports Go applications that run in big data processing scenarios with high concurrency. The library is easy to use and highly customizable. You can use the library to create producers that allow you to resend failed logs. Before Go applications send log data to Log Service, you can use these producers to compress the log data. This improves write performance. For more information about the related GitHub project, visit Aliyun Log Go Producer.

4.3.2.2.5. Python logging

This topic describes how to use the Python logging module to collect log data.

Configurations

For more information about the configurations that are related to the Python logging module, see Logging configuration.

The Python logging module allows you to use code or a configuration file to configure logging. The following example shows how to use the logging.conf configuration file to configure logging.

[loggers] keys=root,sls [handlers] keys=consoleHandler, slsHandler [formatters] keys=simpleFormatter, rawFormatter [logger root] level=DEBUG handlers=consoleHandler [logger sls] level=INFO handlers=consoleHandler, slsHandler qualname=sls propagate=0 [handler consoleHandler] class=StreamHandler level=DEBUG formatter=simpleFormatter args=(sys.stdout,) [handler_slsHandler] class=aliyun.log.QueuedLogHandler level=INFO formatter=rawFormatter args=(os.environ.get('ALIYUN LOG SAMPLE ENDPOINT', ''), os.environ.get('ALIYUN LOG SAMPLE ACCESSID', ''), os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSKEY', ''), os.environ.get('ALIYUN_LOG_SAMPLE_TMP_FROJECT', ''), "logstore") [formatter_simpleFormatter] format=%(asctime)s - %(name)s - %(levelname)s - %(message)s [formatter_rawFormatter]

format=% (message) s

Two handlers named root and sls are created. The sls handler is an object of the aligun.log.QueuedLogHandler class. The following script shows the parameters that you can specify for the sls handler. For more information, see Parameters.

args=(os.environ.get('ALIYUN_LOG_SAMPLE_ENDPOINT', ''), os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSID', ''), os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSKEY', ''), os.environ.get('ALIYUN_LOG_SAMPLE_TMP_PROJECT', ''), "logstore")

? Note

In this case, the os.environ function is used to obtain configurations from environment variables. You can also specify values for these parameters based on your business requirements.

Upload logs

If you want to upload logs to Log Service, you can use the configuration file.

import logging
import logging.config

Configurations
logging.config.fileConfig('logging.conf')
logger = logging.getLogger('sls')

Use the logger
logger.info("test1")

try: 1/0 except ZeroDivisionError as ex: logger.exception(ex)

Then, logs are automatically uploaded to Log Service. If you want to use the query and analysis feature, you must enable the indexing feature for the related Logstore.

Configure indexes for a Logstore

Enable the indexing feature for the Logstore that receives logs and configure indexes for specific fields. We recommend that you use the Log Service command-line interface (CLI) to configure indexes. For more information, see https://www.nython_logging_handler_index.json.

aliyunlog log update_index --project_name="project1" --logstore_name="logstore1" --index_detail="file:///Users/user1/loghandler_index.json"

Specify log fields that you want to collect

The following table describes the log fields that you can collect.

Field	Description
message	The content of a log.
record_name	The name of a handler. In the preceding example, sls is used.
level	The severity level of a log, such as INFO and ERROR.
file_path	The full path of a configuration file.
func_name	The name of a function.
line_no	The number of a log line.
module	The name of a module where the function resides.
thread_id	The ID of the thread that runs the function.
thread_name	The name of the thread that runs the function.
process_id	The ID of the process that runs the function.
process_name	The name of the process that runs the function.

You can specify log fields that you want to collect based on the <u>fields</u> parameter of a class. For more information, see <u>aliyun.log.LogFields</u>. The following example shows how to modify the preceding configuration file and collect several fields, such as module and func_name.

[handler_slsHandler] class=aliyun.log.QueuedLogHandler level=INFO formatter=rawFormatter args=('cn-beijing.log.aliyuncs.com', 'ak_id', 'ak_key', 'project1', "logstore1", 'mytopic', ['level', 'func_name', 'module', 'line_no'])

? Note

- The message field is collected regardless of your configurations.
- If you want to add a prefix and suffix to the names of these fields, use the buildin_fields_prefix and buildin_fields_suffix parameters. Example: __level_ .

Use a JSON text to configure logging

If you want to create flexible logging configurations, you can use a JSON text.
User Guide-Log Service

```
#encoding: utf8
import logging, logging.config, os
# Configurations
conf = {'version': 1,
        'formatters': {'rawformatter': {'class': 'logging.Formatter',
                                           'format': '%(message)s'}
                         },
        'handlers': {'sls_handler': {'()':
                                        'aliyun.log.QueuedLogHandler',
                                        'level': 'INFO',
                                        'formatter': 'rawformatter',
                                        # custom args:
                                        'end_point': os.environ.get('ALIYUN_LOG_SAMPLE_ENDPOINT', ''),
                                        'access_key_id': os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSID', ''),
'access_key': os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSKEY', ''),
                                        'project': 'project1',
                                        'log_store': "logstore1"
                                        }
                      }.
        'loggers': {'sls': {'handlers': ['sls_handler', ],
                                      'level': 'INFO',
                                     'propagate': False}
                     }
       }
logging.config.dictConfig(conf)
# Use the logger
logger = logging.getLogger('sls')
logger.info("Hello world")
```

? Note

If you want to instantiate an object of the aligun.log.QueuedLogHandler class, pass named parameters to the constructor. For more information, see aligun.log.QueuedLogHandler.

4.3.2.3. Collect common logs

4.3.2.3.1. Collect Log4j logs

Log Service allows you to use LogHub Log4j Appender or Logtail to collect Log4j logs.

Log format

Log4j is an open source project of Apache. Log4j allows you to specify the output destination and format of logs. You can also specify the severity level of logs. The severity levels of logs are classified into ERROR, WARN, INFO, and DEBUG in descending order. The output destination specifies whether logs are sent to the console or files. The output format specifies the format of logs. The following example shows the default configurations of Log4j:

```
<Configuration status="WARN">

<Appenders>

<Console name="Console" target="SYSTEM_OUT">

<PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss:SSS zzz} [%t] %-5level %logger{36} - %msg%n"/>

</Console>

</Appenders>

<Logger name="com.foo.Bar" level="trace">

</Logger name="com.foo.Bar" level="trace"</td>
```

The following example shows a sample log:

2013-12-25 19:57:06,954 [10.10.10.10] WARN impl.PermanentTairDaoImpl - Fail to Read Permanent Tair,key:e:470217319319741_1,result:com.example.tair.Result@172e3ebc[rc=code=-1, msg=connection error or timeout,value=,flag=0]

Regular expression that matches each IP address that indicates the start of a line:

\d+-\d+-\d+\s.*

Regular expression that is used to extract log information:

 $(\d+-\d+-\d+\s\d+:\d+,\d+)\s\[([^{]}]*)\]\s\(\S+)\s+(\S+)\s-\s\(.*)$

Time conversion format:

%Y-%m-%d %H:%M:%S

The following table lists the extraction results of the sample log.

Кеу	Value
time	2013-12-25 19:57:06,954
ip	203.0.113.2
level	WARN
class	impl.PermanentTairDaoImpl
message	Fail to Read Permanent Tair,key:e:470217319319741_1,result:com.example.tair.Result@172e3ebc[rc=code=-1, msg=connection error or timeout,value=,flag=0]

Use LogHub Log4j Appender to collect Log4j logs

For more information about how to collect Log4j logs by using LogHub Log4j Appender, see Log4j Appender.

Use Logtail to collect Log4j logs

The procedure when you use Logtail to collect Log4j logs is similar to that when you use Logtail to collect Python logs. Configure Logtail based on the actual network deployment and your business requirements. For more information, see Collect Python logs.

4.3.2.3.2. Collect Python logs

Log Service allows you to use the Python logging module to collect Python logs. This topic describes how to use Logtail to collect Python logs.

Background information

The Python logging module provides a general logging system, which can be used by third-party modules or applications. The logging module defines multiple log severity levels and logging methods. The logging module consists of the following components: loggers, handlers, filters, and formatters. To collect Python logs, we recommend that you use logging handlers. For more information, see the following topics:

- Use logging handlers to automatically upload Python logs
- Use logging handlers to automatically upload and parse logs in the key-value format
- Use logging handlers to automatically parse logs in the JSON format

Log format

Formatters specify the output format of logs. The fields in the configurations of a formatter are in the %(key)s format.

The following table describes the fields in the formatter configurations.

Field	Description
%(name)s	The name of the logger that generates a log.
%(levelno)s	The severity level of a log in the numeric format.
%(levelname)s	The severity level of a log in the text format. Valid values: DEBUG, INFO, WARNING, ERROR, and CRITICAL.
%(pathname)s	The full path name of the source file where the logging call is initiated.
%(filename)s	The name of the source file.
%(module)s	The name of the module where the logging call is initiated.
%(funcName)s	The name of the function from which the logging call is initiated.
%(lineno)d	The line number in the source file where the logging call is initiated.

%(created)f	The time when a log is created. The value is a UNIX timestamp. It is the number of seconds that have elapsed since 00:00:00 UTC, Thursday, January 1, 1970.
%(relativeCreated)d	The difference between the time when a log is created and the time when the logging module is loaded. Unit: milliseconds.
%(asctime)s	The time when a log is created. Example: 2003-07-08 16:49:45,896. The digits after the comma (,) indicate the millisecond portion of the time.
%(msecs)d	The millisecond portion of the time when a log is created.
%(thread)d	The ID of the thread.
%(threadName)s	The name of the thread.
%(process)d	The ID of the process.
%(message)s	The log content.

The following example shows sample logs:

2015-03-04 23:21:59,682 - log_test.py:16 - tst - first info message 2015-03-04 23:21:59,682 - log_test.py:17 - tst - first debug message

Procedure

- 1. Log on to the Log Service console
- 2. In the Import Data section, select RegEx Text Log.
- Select the project and Logstore. Then, click Next.
 You can also click Create Now to create a project and a Logstore.

If you click the plus sign (+) next to **Data Import** below your Logstore to enter the configuration process, the system automatically skips this step. 4. Create a machine group and click **Next**.

- Before you create a machine group, make sure that Logtail is installed. Follow the on-screen instructions to install Logtail. For more information, see Install Logtail in Linux or Install Logtail in Windows. After you install Logtail, click **Complete Installation** to create a machine group. If you created a machine group, click **Use Existing Machine Groups**.
- Select the machine group in the Source Server Groups section and move the machine group to the Applied Server Groups section. Then, click Next.

() Important

If you apply a machine group immediately after you create the machine group, the heartbeat status of the machine group may be **FAIL**. This issue occurs because the machine group is not connected to Simple Log Service. To resolve the issue, you can click **Automatic Retry**. If the status is still **FAIL**, see What do I do if a Logtail machine group has no heartbeats?

Source Server Groups		Applied Server Groups	
Search by server group name	Q	Search by server group name	Q
2	uuud		
	>		
	<		
1 Items		0 Items	

6. Configure the parameters in the Logtail Config step.

i. Configure the Config Name and Log Path parameters and set the Mode parameter to Full Regex Mode.

ii. Turn on Singleline.

iii. Enter a sample log in the Log Sample field.

- iv. Turn on Extract Field.
- v. Specify a regular expression in the $\ensuremath{\textbf{RegEx}}$ field.
- Automatically generate a regular expression.
 - In the Log Sample field, select the content that you want to extract and click Generate Regular Expression. A regular expression is automatically generated.
- Manually enter a regular expression
 Click Manual. In the RegEx field, enter a regular expression. Then, click Validate to check whether the regular expression can be used to parse logs or extract content from logs.
- vi. Verify the result in the **Extracted Content** field.

View the extraction results of log fields and specify keys for the extracted fields.

Specify an informative name for each log field in the extraction results. For example, you can use time as the name for a time field. If you do not use the system time, you must specify the name of a time field in the Value field and time in the Key field.

7. Optional. Configure parameters in the Advanced Options section. After the settings are complete, click Next.

You can configure advanced settings based on your business requirements. We recommend that you retain the default settings. The following table describes the parameters in the Advanced Options section.

Parameter	Description
Enable Plug-in Processing	Specify whether to enable plug-in processing. If you turn on this switch, you can use plug-ins to process text logs. Note If you turn on Enable Plug-in Processing, the parameters such as Upload Raw Log, Timezone, Drop Failed to Parse Logs, Filter Configuration, and Incomplete Entry Upload (Delimiter mode) become unavailable.
Upload Raw Log	If you turn on Upload Raw Log , each raw log is uploaded to Simple Log Service as the value of the_ raw _ field together with the log parsed from the raw log.
Topic Generation Mode	 Select the topic generation mode. Null - Do not generate topic: In this mode, the topic field is set to an empty string. When you query logs, you do not need to specify a topic. This is the default value. Machine Group Topic Attributes: In this mode, topics are configured at the machine group level. If you want to distinguish the logs that are generated by different servers, select this mode. File Path RegEx: In this mode, you must specify a regular expression in theCustom RegEx field. The part of a log path that matches the regular expression is used as the topic. If you want to distinguish the logs that are generated by different users or instances, select this mode.
Custom RegEx	Specify a custom regular expression. If you select File Path RegEx for Topic Generation Mode, you must specify a custom regular expression.
Log File Encoding	Select the encoding format of log files. Valid values: utf8 and gbk. • utf8: UTF-8 encoding is used. • gbk: GBK encoding is used.
Timezone	 Select the time zone for the time of logs that are collected. Valid values: System Timezone: If you select this value, the time zone of the server is used. Custom: If you select this value, you must select a time zone based on your business requirements.
Timeout	 If a log file is not updated within the specified period, Logtail considers the log file to be timed out. Valid values: Never: All log files are continuously monitored, and log files never time out. 30 Minute Timeout: If a log file is not updated within 30 minutes, Logtail considers the log file to be timed out and no longer monitors the file. If you select 30 Minute Timeout, you must configure Maximum Timeout Directory Depth. Valid values: 1 to 3.
Filter Configuration	 Specify the conditions to filter logs. Only logs that exactly match the specified filter conditions are collected. Examples: Collect the logs that match the specified filter conditions. If you setKey to level and set Regex to WARNING/ERROR, only logs in which the value of level is WARNING or ERROR are collected. Filter out the logs that do not match the specified filter conditions. If you set Key to level and set Regex to ^{?!.*(INFO/DEBUG)).*, logs in which the value of level is INFO or DEBUG are filtered out. If you set Key to url and set Regex to .*^ ?!.*(INFO/DEBUG)).*, logs in which the value of url contains healthcheck are filtered out.

8. Configure indexes for query and analysis. After you complete the settings, click **Next**.

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on the collected logs in manual or automatic mode. For more information, see Enable the index feature and configure indexes for a Logstore.

? Note

- If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.
- If you configure an index for a field of the Long or Double type, you cannot configure Case Sensitive or Delimiter for the field.

After you complete the settings, you can start to collect Python logs.

4.3.2.3.3. Collect Node.js logs

Node.js logs are displayed in the Log Service console by default. This affects your data collection and troubleshooting efficiency. Log4js is a tool used to manage Node.js logs. You can use Log4js to send Node.js logs to files and customize the log format. Log4js allows you to collect and consolidate data in an efficient manner.

The following code shows how to configure Log4js to send logs to a file:

```
var log4js = require('log4js');
log4js.configure({
  appenders: [
    {
     type: 'file', // Output to a file
     filename: 'logs/access.log',
     maxLogSize: 1024,
     backups:3,
     category: 'normal'
   }
 1
});
var logger = log4js.getLogger('normal');
logger.setLevel('INFO');
logger.info("this is a info msg");
logger.error("this is a err msg");
```

Log format

After you use Log4js to write logs to text files, the logs are displayed in the following format:

[2016-02-24 17:42:38.946] [INFO] normal - this is a info msg [2016-02-24 17:42:38.951] [ERROR] normal - this is a err msg

Log4js classifies log severities into the following six levels in ascending order: TRACE, DEBUG, INFO, WARN, ERROR, and FATAL.

Use Logtail to collect Node.js logs

The procedure when you configure Logtail to collect Node.js logs is similar to that when you configure Logtail to collect Python logs. For more information, see Collect Python logs. Set related parameters based on the actual network deployment and your business requirements. The regular expression that is automatically generated is based on the sample log and may not apply to other logs. Therefore, you must modify the regular expression based on your business requirements before you use it. You can use the following sample Node.js logs to configure regular expressions for your logs.

Sample Node.js logs and regular expressions:

- Example 1
- Sample log

[2016-02-24 17:42:38.946] [INFO] normal - this is a info msg

Regular expression

 $[([^]]+)] s [([^]]+)] s (w+) s-(.*)$

Extracted fields

time , level , loggerName , and message

Example 2

Sample log

[2016-01-31 12:02:25.844] [INFO] access - 42.120.73.203 - - "GET /user/projects/ali_sls_log?ignoreError=true HTTP/1.1" 304 - "http:// aliyun.com/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.97 Safari/537.36"

Regular expression

Extracted fields

time , level , loggerName , ip, request , status , referer , and user_agent

4.3.2.3.4. Collect WordPress logs

This topic describes the format of WordPress logs and extraction results of a sample log.

Log format

Sample log:

172.64.0.2 - [07/Jan/2016:21:06:39 +0800] "GET /wp-admin/js/password-strength-meter.min.js?ver=4.4 HTTP/1.0" 200 776 "http://wordpress.c4a1a0aecdb1943169555231dcc4adfb7.cn-hangzhou.alicontainer.com/wp-admin/install.php" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36"

Configure Logtail to collect WordPress logs

- If you use Logtail to collect WordPress logs, you must configure the following settings:
- Regular expression that matches each IP address that indicates the start of a line

\d+\.\d+\.\d+\.\d+\s-\s.*

• Regular expression that is used to extract log information

• Time conversion format

%d/%b/%Y:%H:%M:%S

The following table lists the extraction results of the sample log.

Кеу	Value
ip	10.10.10.1
time	07/Jan/2016:21:06:39 +0800
method	GET
url	/wp-admin/js/password-strength-meter.min.js?ver=4.4 HTTP/1.0
status	200
length	776
ref	http://wordpress.c4a1a0aecdb1943169555231dcc4adfb7.cn-hangzhou.alicontainer.com/wp-admin/install.php
user-agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36

4.3.2.3.5. Collect Unity3D logs

This topic describes how to use the web tracking feature of Log Service to collect Unity3D logs.

Background information

Unity3D is a cross-platform game engine that is developed by Unity Technologies. You can use the engine to create 3D video games, VR buildings, real-time 3D animation, and other interactive content.

In this example, Unity Debug.Log is used to describe how to collect Unity3D logs.

Procedure

- 1. Enable the web tracking feature.
- For more information, see Use the web tracking feature to collect logs.
- Create a Unity3D logging handler. In the Unity editor, create a C# file named LogOutputHandler.cs, add the following code to the file, and then modify the following variables: • project: the name of the Log Service project.
 - logstore: the name of the Logstore.
 - serviceAddr: the endpoint of the Log Service project. For more information, see Obtain an endpoint in Log Service Developer Guide.

```
using UnityEngine;
using System.Collections;
public class LogOutputHandler : MonoBehaviour
    //Register the HandleLog function on scene start to fire on debug.log events
    public void OnEnable()
       Application.logMessageReceived += HandleLog;
    //Remove callback when object goes out of scope
    public void OnDisable()
       Application.logMessageReceived -= HandleLog;
    string project = "your project name";
    string logstore = "your logstore name";
    string serviceAddr = "http address of your log service project";
    //Capture debug.log output, send logs to Loggly
    public void HandleLog(string logString, string stackTrace, LogType type)
        string parameters = "";
       parameters += "Level=" + WWW.EscapeURL(type.ToString());
       parameters += "&";
       parameters += "Message=" + WWW.EscapeURL(logString);
       parameters += "&";
       parameters += "Stack_Trace=" + WWW.EscapeURL(stackTrace);
       parameters += "&";
       //Add any User, Game, or Device MetaData that would be useful to finding issues later
       parameters += "Device_Model=" + WWW.EscapeURL(SystemInfo.deviceModel);
        string url = "http://" + project + "." + serviceAddr + "/logstores/" + logstore + "/track?APIVersion=0.6.0%" + parameters;
       StartCoroutine(SendData(url));
    public IEnumerator SendData(string url)
         NWW sendLog = new WWW(url);
       yield return sendLog;
    }
}
```

You can use the preceding code to asynchronously send logs to Log Service. You can also specify other fields in the code to collect the fields. 3 Generate Unitv3D logs

Create a file named LogglyTest.cs and add the following code to the file:

```
using UnityEngine;
using System.Collections;
public class LogglyTest : MonoBehaviour {
   void Start () {
       Debug.Log ("Hello world");
}
```

View logs in the Log Service console. After you run the Unity3D application, logs are generated and sent to Log Service. You can view the logs in the Log Service console.

4.3.2.4. Data import

4.3.2.4.1. Import data from OSS to Simple Log Service

You can upload log files to Object Storage Service (OSS) buckets for storage. Then, you can import the log data from OSS to Simple Log Service and perform supported operations on the data in Simple Log Service. For example, you can query, analyze, and transform the data. You can import only the OSS objects that are no more than 5 GB in size to Simple Log Service. If you want to import a compressed object, the size of the compressed object must be no more than 5 GB.

Prerequisites

- Log files are uploaded to an OSS bucket. For more information, see Upload objects in OSS User Guide > Quick start.
- A project and a Logstore are created. For more information, see Create a project and Create a Logstore.
- Simple Log Service is authorized to assume the AliyunLogImportOSSRole role to access your OSS resources.

If you use a RAM user, you must grant the PassRole permission to the RAM user. The following example shows a policy that you can use to grant the PassRole permission. For more information, see Use custom policies to grant permissions to a RAM user and Permissions required by a RAM user to manage Simple Log Service res

```
"Statement": [
  "Effect": "Allow",
   "Action": "ram:PassRole",
   "Resource": "acs:ram:*:*:role/aliyunlogimportossrole"
],
 "Version": "1"
}
```

Create a data import configuration

() Important

Data import jobs import full data of updated OSS objects to Simple Log Service. If new data is appended to an OSS object that is imported to Simple Log Service, all data of the OSS object is re-imported to Simple Log Service when a data import job for the OSS object is run.

1. Log on to the Simple Log Service console

- 2. On the Data Import tab in the Import Data section, click OSS Data Import .
- 3. Select the project and Logstore. Then, click **Next**.
- 4. In the Configure Import Settings step, create a data import configuration.

i. In the **Configure Import Settings** step, configure the parameters. The following table describes the parameters.

Parameter	Description		
Config Name	The name of the data import configuration.		
OSS Region	The region where the OSS bucket resides. The OSS bucket stores the OSS objects that you want to import to Simple Log Service. If the OSS bucket and the Simple Log Service project reside in the same region, no Internet traffic is generated, and data is transferred at a high speed.		
Bucket	The OSS bucket.		
File Path Prefix Filter	The directory of the OSS objects. If you configure this parameter, the system can find the OSS objects that you want to import in a more efficient manner. For example, if the OSS objects that you want to import are stored in the csv/directory, you can set this parameter to csv/. If you leave this parameter empty, the system traverses the entire OSS bucket to find the OSS objects. Note We recommend that you configure this parameter. The larger the number of OSS objects in an OSS bucket is, the lower the data import efficiency becomes when the entire bucket is traversed.		
File Path Regex Filter	The regular expression that is used to filter OSS objects by directory. If you configure this parameter, the system can find the OSS objects that you want to import in a more efficient manner. Only the objects whose names match the regular expression are imported. The names include the paths of the objects. By default, this parameter is left empty, which indicates that no filtering is performed. For example, if an OSS object that you want to import is namedtestdata/csv/bill.csv, you can set this parameter to (testdata/csv/)(.*) . For information about how to test a regular expression, seeHow do I test a regular expression?		
Data Format	 The format of the OSS objects. Valid values: CSV: You can specify the first line of an OSS object as field names or specify custom field names. All lines except the first line are parsed as the values of log fields. Single-line JSON: An OSS object is read line by line. Each line is parsed as a JSON object. The fields in JSON objects are log fields. Single-line Text: Each line in an OSS object are parsed as a log. Multi-line Text: Multiple lines in an OSS object are parsed as a log. You can specify a regular expression to match the first line or the last line in a log. Alibaba Cloud OSS Access Log: OSS data is parsed based on the format of access logs of Alibaba Cloud OSS. For more information, see <i>Configure logging in OSS User Guide > Buckets</i>. Alibaba Cloud CDN Download Log: OSS data is parsed based on the format of download logs of Alibaba Cloud CDN (CDN). ORC: An OSS object in the Optimized Row Columnar (ORC) format is automatically parsed into the format that is supported by Simple Log Service without manual configurations. Parquet: An OSS object in the Parquet format is automatically parsed into the format that is supported by Simple Log Service without manual configurations. 		
Compression Format	The compression format of the OSS objects that you want to import. Simple Log Service decompresses the OSS objects based on the specified format to read data.		
Encoding Format	The encoding format of the OSS objects that you want to import. Only UTF-8 and GBK are supported.		
Import Archive Files	 If the OSS objects that you want to import are of the Archive or Cold Archive storage class, Simple Log Service can read data from the objects only after the objects are restored. If you turn on this switch, Archive and Cold Archive objects are automatically restored. Note Restoring Archive objects requires approximately 1 minute, which may cause the first preview to time out. If the first preview times out, try again later. Restoring Cold Archive objects requires approximately 1 hour. If the preview times out, you can skip the preview or try again 1 hour later. By default, restored Cold Archive objects are valid within seven days. This way, the system has sufficient time to import Cold Archive objects to Simple Log Service. 		
Use System Time	If the time fails to be parsed, the current time is used as the log time.		

ii. Confirm the settings and click Next.

5. Preview data, configure indexes, and then click Next.

By default, full-text indexing is enabled in Simple Log Service. You can configure field indexes based on collected logs in manual or automatic mode. To configure field indexes in automatic mode, click **Automatic Index Generation**. Simple Log Service automatically creates field indexes. For more information, see Configure indexes.

() Important

If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable full-text indexing and field indexing, the system uses only field indexes.

6. Click Log Query. On the query and analysis page, check whether OSS data is imported.

Wait for approximately 1 minute. If the required OSS data is imported, the import is successful.

View a data import configuration

After you create a data import configuration, you can view the configuration details and related reports in the Simple Log Service console.

1. In the **Projects** section, click the project to which the data import configuration belongs.

 In the left-side navigation pane, choose Log Storage > Logstores. Click the Logstore to which the data import configuration belongs, choose Data Import > Data Import, and then click the name of the data import configuration.

3. On the Import Configuration Overview page, view the basic information about the data import configuration and the related reports.

What to do next

On the Import Configuration Overview page, you can perform the following operations on the data import configuration:

Modify the data import configuration

To modify the data import configuration, click **Modify Settings**. For more information, see Create a data import configuration. • Delete the data import configuration

To delete the data import configuration, click **Delete Configuration**.

🔥 Warning

After a data import configuration is deleted, it cannot be restored. Proceed with caution.

FAQ

Issue	Cause	Solution
Garbled characters exist.	The data format, compression format, or encoding format is not configured as expected.	Check the actual format of the OSS objects, and then modify the Data Format , Compression Format , or Encoding Format parameter. To handle the existing garbled characters, create a Logstore and a data import configuration.
The log time displayed in Simple Log Service is different from the actual time of the imported data.	The time field is not specified in the data import configuration, or the specified time format or time zone is invalid.	Specify a time field or specify a valid time format or time zone. For more information, see Create a data import configuration.
After data is imported, the data cannot be queried or analyzed.	 The data is not within the query time range. No indexes are configured. The indexes failed to take effect. 	 Check whether the time of the data that you want to query is within the query time range that you specify. If not, adjust the query time range and query the data again. Check whether indexes are configured for the Logstore to which the data is imported. If not, configure indexes first. For more information, see Configure indexes. If indexes are configured for the Logstore and an expected volume of imported data is displayed on the Data Processing Insight dashboard, the possible cause is that the indexes failed to take effect. In this case, reindex the data.
The number of imported data entries is less than expected.	Some OSS objects contain data in which a line is greater than 3 MB in size. In this case, the data is discarded during the import.	When you write data to an OSS object, make sure that the size of a line does not exceed 3 MB.
The number of OSS objects and the total volume of data are large, but the import speed does not meet your expectation. In most cases, the import speed can reach 80 MB/s.	The number of shards in the Logstore is excessively small.	If the number of shards in a Logstore is small, increase the number of shards to 10 or more and check the latency.
You cannot select an OSS bucket when you create a data import configuration.	The AliyunLogImportOSSRole role is not assigned to Simple Log Service.	Complete authorization. For more information, see the "Prerequisites" section of this topic.

Cloud Defined Storage

Some OSS objects failed to be imported to Simple Log Service.	The settings of the filter conditions are invalid or the size of a single object exceeds 5 GB.	 Check whether the OSS objects that you want to import meet the filter conditions. If the objects cannot meet the filter conditions, modify the filter conditions. Make sure that the size of each OSS object that you want to import is less than 5 GB. If the size of an object exceeds 5 GB, reduce the size of the object.
No Archive objects are imported to Simple Log Service.	Import Archive Files is turned off.	 Method 1: Modify the data import configuration and turn on Import Archive Files. Method 2: Create a data import configuration and turn on Restore Archived Files.
Multi-line text logs are incorrectly parsed.	The specified regular expression that is used to match the first line or the last line in a log is invalid.	Check whether the regular expression that is used to match the first line or the last line in a log is valid.
The latency to import new OSS objects is higher than expected.	The number of existing OSS objects that meet the conditions specified by File Path Prefix Filter exceeds the limit and OSS Metadata Indexing is turned off in the data import configuration.	If the number of existing OSS objects that meet the conditions specified by File Path Prefix Filter exceeds one million, turn on OSS Metadata Indexing in the data import configuration. Otherwise, the efficiency of new file discovery is low.

Error handling

Item	Description
File read failure	If an OSS object fails to be completely read because a network exception occurs or the object is damaged, the corresponding data import job automatically retries to read the object. If the object fails to be read after three retries, the object is skipped. The retry interval is the same as the value of the New File Check Cycle parameter. If the New File Check Cycle parameter is set to Never Check, the retry interval is 5 minutes.
Compression format parsing error	If the compression format is invalid when an OSS object is decompressed, the corresponding data import job skips the object.
Data format parsing error	 If data in the binary format (ORC or Parquet) fails to be parsed, the corresponding data import job skips the OSS object If data in other formats fails to be parsed, the data import job stores the original text content in the content field of logs.
Logstore not exist	A data import job periodically retries. The data import job does not resume the import until the Logstore is recreated. If the Logstore does not exist, the data import job does not skip any OSS objects. Therefore, after the Logstore is recreated, the data import job automatically imports data from the unprocessed objects in the OSS bucket to the Simple Log Service Logstore.
OSS bucket not exist	A data import job periodically retries. The data import job does not resume the import until the OSS bucket is recreated.
Permission error	If a permission error occurs when data is read from an OSS bucket or data is written to a Simple Log Service Logstore, the corresponding data import job periodically retries. The data import job does not resume the import until the error is fixed. If a permission error occurs, the data import job does not skip any OSS objects. Therefore, after the error is fixed, the data import job automatically imports data from the unprocessed objects in the OSS bucket to the Simple Log Service Logstore.

4.3.2.4.2. Time formats

When you create an import task, you must specify a format for the time field. This topic describes the syntax of time formats and provides examples of time formats.

Time format syntax

Character	Description	Example
G	Era designator	AD
у	Year	2001
М	Month	July or 07
d	Day	10
h	Hour in AM or PM (1 to 12)	12

н	Hour of the day (0 to 23)	22
m	Minute	30
S	Second	55
S	Millisecond	234
E	Week	Tuesday
D	Day of the year	360
F	Day of the week in month	2
w	Week of the year	40
W	Week of the month	1
а	AM or PM	РМ
k	Hour of the day (1 to 24)	24
К	Hour in AM or PM (0 to 11)	10
Z	Time zone	Eastern Standard Time
1	Delimiter	Delimiter
н	Single quotation marks	н

Time format examples

Date format	Parsing syntax	Parsed value (unit: seconds)
2020-05-02 17:30:30	yyyy-MM-dd HH:mm:ss	1588411830
2020-05-02 17:30:30:123	yyyy-MM-dd HH:mm:ss:SSS	1588411830
2020-05-02 17:30	yyyy-MM-dd HH:mm	1588411800
2020-05-02 17	yyyy-MM-dd HH	1588410000
20-05-02 17:30:30	yy-MM-dd HH:mm:ss	1588411830
2020-05-02T17:30:30Z	yyyy-MM-dd'T'HH:mm:ss'Z'	1588411830
2020-05-02T17:30:30.387Z	yyyy-MM-dd'T'HH:mm:ss.SSS'Z'	1588411830
02/May/2020:17:30:30 +0800	dd/MMM/yyyy:HH:mm:ss XX	1588411830
02/May/2020:17:30:30 +08:00	dd/MMM/yyyy:HH:mm:ss XXX	1588411830
2/May/2020:17:30:30 +0800	d/MMM/yyyy:HH:mm:ss XX	1588411830
Sat May 02 17:30:30 CST 2020	EEE MMM dd HH:mm:ss zzz yyyy	1588411830

4.4. Query and analysis

4.4.1. Log search overview

Log Service allows you to search billions to hundreds of billions of logs in seconds.

Syntax

Each query statement consists of a search statement and an analytic statement. The search statement and the analytic statement are separated with a vertical bar (]). The syntax of search statements is used only in Log Service. For more information, see Search syntax.

? Note

- A search statement can be executed alone. However, an analytic statement must be executed together with a search statement. You can use the log analysis feature to analyze the data in search results. You can also use the feature to analyze all data in a Logstore.
- If you want to search tens of billions of logs, you can repeatedly execute a search statement up to 10 times to obtain the complete result.

Syntax

Search statement|Analytic statement

Statement	Description
Search statement	A search statement specifies one or more search conditions and returns the logs that meet the specified conditions. A search statement can be a keyword, a numeric value, a numeric value range, a space, or an asterisk (*). If you specify a space or an asterisk (*) as the search statement, no conditions are used for searching, and all logs are returned. For more information, see Search syntax.
Analytic statement	An analytic statement is used to aggregate and compute the data in search results or all data in a Logstore. For more information, see Log analysis overview.

• Example

* | SELECT status, count(*) AS PV GROUP BY status

Limits

Item	Description	Remarks
Number of keywords	The number of keywords that are used as search conditions. The number of logical operators is not included. You can specify up to 30 keywords in a search statement.	None.
Size of a field value	The maximum size of a field value is 10 KB. The excess part is not involved in searching.	If the size of a field value is greater than 10 KB, logs may fail to be obtained by using keywords, but the logs are actually stored in the Logstore.
Maximum number of concurrent search statements	Each project supports up to 100 concurrent search statements.	For example, 100 users can concurrently execute search statements in all Logstores of a project.
Returned result	The returned logs are displayed on multiple pages. Each page displays up to 100 logs.	None.
Maximum size of a log	Log Service performs the Document Object Model (DOM) operation only on the first 10,000 characters of a log due to browser performance limits.	If a log contains more than 10,000 characters, the following message appears in the Log Service console: The log contains log data of more than 10,000 characters, and some display will be downgraded.
Fuzzy search	In a fuzzy search, Log Service matches up to 100 words that meet the specified conditions and returns the logs that meet the search conditions and contain one or more of these words. For more information, see Fuzzy search.	None.
Data sorting in search results	By default, search results are displayed in descending order of time, which is accurate to minutes.	None

Operation methods

() Important

Before you search logs, make sure that logs are collected and indexes are configured. Indexes are used in a storage structure to sort one or more columns of log data. For more information, see Configure indexes.

Use the Log Service console

Log on to the Log Service console. On the query and analysis page of a Logstore, specify a time range and a search statement to query logs. For more information, see Query and analyze logs and Search syntax.

Call API operations

Call API operations to query logs. For more information, see GetLogs and GetHistograms in the API Reference topic of Developer Guide.

4.4.2. Log analysis overview

Log Service provides the log analysis feature. This feature allows you to search for log data and use SQL functions to analyze the data. This topic describes the syntax and limits of the analytic statements. This topic also provides the SQL functions that you can call when you use the log analysis feature.

? Note

If you want to use the log analysis feature, you must turn on **Enable Analytics** when you configure indexes for log fields. For more information, see **Configure indexes**. If you turn on **Enable Analytics**, you can analyze log data within seconds without additional costs.

Syntax

Each query statement consists of a search statement and an analytic statement. The search statement and the analytic statement are separated by a vertical bar (]). You can execute a search statement alone. However, you must execute an analytic statement together with a search statement. You can use the log analysis feature to analyze data that meets specified search conditions in a Logstore. You can also use the feature to analyze all data in a Logstore.

? Note

- You do not need to specify a FROM or WHERE clause in an analytic statement. By default, all data of the current Logstore is analyzed.
- You do not need to add a semicolon (;) at the end of an analytic statement to end the statement.
- · Analytic statements are case-insensitive.
- Syntax

Search statement | Analytic statement

Statement	Description
Search statement	A search statement specifies one or more search conditions. A search statement can be a keyword, a value, a value range, a space character, or an asterisk (*). If you specify a space character or an asterisk (*) as the search statement, no conditions are specified and all logs are returned. For more information, see Search syntax.
Analytic statement	An analytic statement is used to aggregate or analyze all log data or the log data that meets the specified search conditions in a Logstore.

Example

 \star | SELECT status, count(*) AS PV GROUP BY status

Limits

- Each project supports a maximum of 15 concurrent analytic statements at the same time.
- For example, 15 users can concurrently execute analytic statements in all Logstores of a project at the same time.
- You can analyze only the data that is written to Log Service after the log analysis feature is enabled.
- By default, an analytic statement returns a maximum of 100 rows of data.
- If you want to view more data, use a LIMIT clause. For more information, see LIMIT syntax.
- The maximum size of a field value is 16 KB. If the size of a field value exceeds 16 KB, the excess content is not analyzed.
- The maximum timeout period for an analytic statement is 55 seconds.
- Each shard supports only 1 GB of data for an analytic statement.
- The value of a double-type field can contain a maximum of 52 digits after the decimal point.
- If the number of digits after the decimal point is greater than 52, the accuracy of the field value is compromised.

SQL functions and syntax

This section lists the SQL functions and syntax that Log Service supports.

- The following aggregate functions are available for SELECT statements:
- General aggregate functions
- Security check functions
- Map functions and operators
- Approximate functions
- Mathematical statistics functions
- Mathematical calculation functions
- String functions
- Date and time functions
- URL functions
- Regular expression functions
- JSON functions
- Type conversion functions
- IP functions
- Array functions and operators
- Binary string functions
- Bitwise functions
- Interval-valued comparison and periodicity-valued comparison functions
- Comparison functions and operators
- Lambda expressions
- Logical functions

Cloud Defined Storage

- Geospatial functions
- Geography functions
- Machine learning syntax and functions
- GROUP BY clause
- Window functions
- HAVING clause
- ORDER BY clause
- LIMIT syntax
- Conditional expressions
- UNNEST clause
- Column aliases
- Nested subquery

4.4.3. Reserved fields

This topic describes the reserved fields of Log Service.

! Important

- When you call API operations to write data or create Logtail configurations, we recommend that you do not use the names of the reserved fields as field names in the operations. Otherwise, issues such as duplicate field names and inaccurate queries may occur.
- The fields that have the ______ prefix cannot be shipped.

Reserved field	Data format	Index and log analysis configuration	Description	
time	The value is an integer that represents a UNIX timestamp.	 Index configuration: Thetime field is specified by using the from and to parameters in API operations. You do not need to create an index for thetime field. Log analysis configuration: By default, if you turn on Enable Analytics for any field, the log analysis feature is enabled for thetime field. 	The time when a log is written to a Logstore. You can use this field to ship, query, and analyze logs.	
Source	The value is a string.	 Index configuration: By default, if you enable the indexing feature, Log Service creates an index for thesource field. The index is of the text type. No delimiter is specified for the index. To query logs based on the index, you can enter source:127.0.0.1 orsource:127.0.0.1 Log analysis configuration: By default, if you turn on Enable Analytics for any field, the log analysis feature is enabled for thesource field. 	The machine from which logs are collected. You can use this field to ship, query, analyze, and consume logs.	
topic	The value is a string.	 Index configuration: By default, if you enable the indexing feature, Log Service creates an index for thetopic field. The index is of the text type. No delimiter is specified for the index. To query loas based on the index, you can entertopic:XXX Log analysis configuration: By default, if you turn on Enable Analytics for any field, the log analysis feature is enabled for thetopic field. 	The topic of a log. If you specify a topic for a log, Log Service adds a topic field to the log. The key of the field is <u>topic</u> , and the value of the field is the topic content. You can use this field to ship, query, analyze, and consume logs. For more information, see Topic.	
_extract_others_	The value is a string and can be deserialized into a JSON map.	This field does not exist in logs. You do not need to create an index for this field.	This field is equivalent to the extract_others field. We recommend that you use theextract_others field.	
tag:client_ip_ _	The value is a string.	 Index configuration: By default, if you enable the indexing feature, Log Service creates an index for the tag field. The index is of the text type. No delimiter is specified for the index. Exact match and fuzzy match are supported for log queries. Log analysis configuration: By default, the log analysis feature is disabled for the _tag_: client_ipfield. To enable the log analysis feature for the _ index_ip field, you must create an index for the field and turn on Enable Analytics for the field. 	The public IP address of the machine from which logs are collected. This field is a system tag. If you enable the public IP address recording feature, this field is added to each raw log when Log Service receives logs. You can use this field to query, analyze, and consume logs. When you specify this field in an SQL statement, you must enclose this field in double quotation marks (""). For more information, see Log and Manage Logstores.	

User Guide-Log Service

Cloud Defined Storage

tag:_receive_ti me	The value is string and can be converted into an integer that represents a UNIX timestamp.	 Index configuration: By default, if you enable the indexing feature, Log Service creates an index for the tag field. The index is of the text type. No delimiter is specified for the index. Exact match and fuzzy match are supported for log queries. Log analysis configuration: By default, the log analysis feature is disabled for thetag_:_receive_time field. To enable the log analysis feature for thetag_:_receive_time field, you must create an index for the field and turn on Enable Analytics for the field. 	The time when Log Service receives a log. This field is a system tag. If you enable the public IP address recording feature, this field is added to each raw log when Log Service receives logs. You can use this field to query, analyze, and consume logs. For more information, see Log and Manage Logstores.
tag_:_path	The value is a string.	 Index configuration: By default, if you enable the indexing feature, Log Service creates an index for thetag_:_path field. The index is of the text type. No delimiter is specified for the index. To query loas based on the index, you can entertag_:_path:XXX . Log analysis configuration: By default, the log analysis feature is disabled for the _tag_:_path field. To enable the log analysis feature for thetag_:_path field, you must create an index for the field and turn on Enable Analytics for the field. 	The path to the log file from which logs are collected. Logtail automatically adds this field to the collected logs. You can use this field to query, analyze, and consume logs. When you specify this field in an SQL statement, you must enclose this field in double quotation marks ("").
tag_:_hostname	The value is a string.	 Index configuration: By default, if you enable the indexing feature, Log Service creates an index for the <u>tag</u>:<u>hostname</u> field. The index is of the text type. No delimiter is specified for the index. To query logs based on the index, you can enter <u>tag</u>:<u>hostname</u>:XXX . Log analysis configuration: By default, the log analysis feature is disabled for the <u>tag</u>:<u>hostname</u>, field. To enable the log analysis feature for the <u>tag</u>:<u>hostname</u>	The hostname of the machine from which Logtail collects logs. Logtail automatically adds this field to logs. You can use this field to query, analyze, and consume logs. When you specify this field in an SQL statement, you must enclose this field in double quotation marks ("").
raw_log	The value is a string.	You must create and configure an index of the text type for this field and enable the log analysis feature based on your business requirements.	The raw log that fails to be parsed. If you turn off Drop Failed to Parse Logs, Logtail uploads raw logs that fail to be parsed. The key of this field is raw_log, and the value of this field is the log content. You can use this field to ship, query, analyze, and consume logs.
raw	The value is a string.	You must create and configure an index of the text type for this field and enable the log analysis feature based on your business requirements.	The raw log that is parsed. If you turn on Upload Raw Log, Logtail uploads the raw logs as the $\{raw}_$ field together with the parsed logs. You can use this field in log audit and compliance check scenarios. You can use this field to ship, query, analyze, and consume logs.

4.4.4. Configure indexes

An index is a storage structure used to sort one or more columns of log data. You can query and analyze log data only after you configure indexes. Query and analysis results vary based on index configurations. Therefore, you must configure indexes based on your business requirements.

Prerequisites

Logs are collected. For more information, see Data collection overview.

Index types

The following table describes the index types supported by Log Service.

Index type	Description
Full-text index	Log Service splits an entire log into multiple words based on specified delimiters to create indexes. In a search statement, the field names (keys) and field values (values) are both plain text. For example, the search statement error returns the logs that contain the keyword error.
Field index	After you configure field indexes, you can specify field names and field values in the Key:Value format to search for logs. For example, the search statement level:error returns the logs whose level field value contains error. If you want to use the analysis feature, you must configure field indexes and turn on Enable Analytics for the required fields. If you turn on Enable Analytics, no additional index traffic is generated, and no additional storage space is occupied.

! Important

- The indexing feature takes effect only on the log data that is written after you configure indexes.
- If you configure both full-text indexes and field indexes, the configurations of the field indexes take precedence.

Configure full-text indexes

1. Log on to the Simple Log Service console

- 2. In the Projects section, click the project that you want to manage.
- 3. On the Log Storage > Logstores tab, click the Logstore that you want to manage.
- 4. On the page that appears, choose Index Attributes > Attributes.
- If the indexing feature is not enabled, click **Enable**.
- 5. Configure indexes.

? Note

If a Logstore is a dedicated Logstore for a cloud service or an internal Logstore, you should turn off Auto Update, then you can configure indexex.

Parameter	Description		
LogReduce	If you turn on LogReduce , Log Service automatically aggregates text logs that have the same pattern during log collection. This way, you can obtain the overall information about logs. For more information, see LogReduce.		
Full Text Index	If you turn on Full Text Index , the full-text indexing feature is enabled.		
Case Sensitive	 Specifies whether searches are case-sensitive. If you turn on Case Sensitive, searches are case-sensitive. For example, if a log contains internalError, you can search for the log by using only the keyword internalError. If you turn off Case Sensitive, searches are not case-sensitive. For example, if a log contains internalError, you can search for the log by using the keyword INTERNALERROR or internalerror. 		
Include Chinese	Specifies whether to distinguish between Chinese content and English content in searches. • After you turn on Include Chinese, if a log contains Chinese characters, the Chinese content is split based on the Chinese grammar. The English content is split based on specified delimiters. ① Important When the Chinese content is split, the write speed is reduced. Proceed with caution. • If you turn off Include Chinese, all the content in a log is split based on specified delimiters.		
Delimiter	 The delimiters that are used to split the content of a log into multiple words. Supported delimiters include , '";=()[]{? %4<>/:\n\t\r . \n indicates a line feed, \t indicates a tab character, and \r indicates a carriage return. For example, the content of a log is /url/pic/abc.gif . o If you do not specify a delimiter, the log is regarded as a single word /url/pic/abc.gif . You can search for the log only by using the keyword /url/pic/abc.gif or by using /url/pic/* to perform a fuzzy search. o If you set Delimiter to a forward slash (/), the content of the log is split into the following three words url , pic , and abc.gif . You can search for the log by using the keyword url , abc.gif , or /url/pic/abc.gif , or by using pi* to perform a fuzzy search. o If you set Delimiter to a forward slash (/) and a period (.), the content of the log is split into the following four words: url , pic , abc , and gif . 		

6. Click **OK**.

The index configurations take effect within 1 minute.

Configure field indexes

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. On the Log Storage > Logstores tab, click the Logstore that you want to manage.
- 4. On the page that appears, choose Index Attributes > Attributes.
- If the indexing feature is not enabled, click **Enable**.
- 5. On the **query and analysis** page, configure indexes.

? Note

If a Logstore is a dedicated Logstore for a cloud service or an internal Logstore, you should turn off Auto Update, then you can configure	
indexex.	

Parameter

Description

	The name of the log field. Example: client_ip.
Key Name	 Note If you configure an index for a tag field, you must set the Key Name parameter in the _tag_:KEY format. For example, you can set the parameter to _tag :_receive_time Different tag fields are supported. For example, a tag field that indicates a public IP address or a UNIX timestamp is supported. When you configure an index for a tag field, numeric data types are not supported. You must set the Type parameter for each tag field to text.
Туре	The data type of the log field value. Valid values: text, long, double, and json. For more information, se@ata types. Note If you set the data type for a field to long or double, you cannot configure theCase Sensitive, Include Chinese, or Delimiter parameter for the field.
Alias	The alias of the field. Example: ip . An alias is used only in analytic statements. You must use the original field name in search statements. For more information, see Column aliases.
Case Sensitive	 Specifies whether searches are case-sensitive. If you turn on Case Sensitive, searches are case-sensitive. For example, if a log contains internalError, you can search for the log by using only the keyword internalError. If you turn off Case Sensitive, searches are not case-sensitive. For example, if a log contains internalError, you can search for the log by using the keyword INTERNALERROR or internalerror.
Delimiter	 The delimiters that are used to split the content of a log into multiple words. Supported delimiters include , '";=()[]{? %i<>/:\n\t\r . \n indicates a line feed, \t indicates a tab character, and \r indicates a carriage return. For example, the content of a log is /url/pic/abc.gif . If you do not specify a delimiter, the log is regarded as a single word /url/pic/abc.gif . You can search for the log only by using the keyword /url/pic/abc.gif or by using /url/pic/* to perform a fuzzy search. If you set Delimiter to a forward slash (/), the content of the log is split into the following three words url , pic , and abc.gif . You can search for the log by using the keyword url , abc.gif , or /url/pic/abc.gif , or by using pi* to perform a fuzzy search. If you set Delimiter to a forward slash (/) and a period (.), the content of the log is split into the following four words: url , pic , abc , and gif .
Include Chinese	 Specifies whether to distinguish between Chinese content and English content in searches. After you turn on Include Chinese, if a log contains Chinese characters, the Chinese content is split based on the Chinese grammar. The English content is split based on specified delimiters. Important When the Chinese content is split, the write speed is reduced. Proceed with caution. If you turn off Include Chinese, all the content in a log is split based on specified delimiters.
Enable Analytics	Before you can use the analysis feature, you must turn on Enable Analytics .

6. Click **OK**.

The index configurations take effect within 1 minute.

4.4.5. Query and analyze logs

After you enable the indexing feature and configure indexes for a Logstore, you can query and analyze the logs that are stored in the Logstore in real time.

Prerequisites

- Logs are collected and stored in a Logstore. For more information, see Data collection.
- The indexing feature is enabled and indexes are configured. For more information, see Configure indexes.

Query and analyze logs

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project to which the Logstore belongs.
- 3. Choose Log Storage > Logstores. On the Logstores tab, click the Logstore where logs are stored.
- 4. Enter a query statement in the search box.

A query statement consists of a search statement and an analytic statement in the Search statement|Analytic statement format. For more information, see Search syntax and SQL syntax and functions.

5. Click **15 Minutes(Relative)** to specify a time range.

You can select a relative time or a time frame. You can also specify a custom time range.

? Note

The query results may contain logs that are generated 1 minute earlier or later than the specified time range.

6. Click Search & Analyze to view the query and analysis results.

Manage query and analysis results

You can view the query and analysis results in a log distribution histogram, on the Raw Logs tab, or in a chart that is displayed on the Graph tab. You can also configure alerts and saved searches.

? Note

By default, only 100 rows of data are returned after you execute a query statement. You can use a LIMIT clause to change the number of returned rows. For more information, see LIMIT syntax.

- · Log distribution histogram
 - The log distribution histogram displays the distribution of query and analysis results in different time ranges.
 - If you move the pointer over a green rectangle, you can view the time range that is represented by the rectangle and the number of logs that are
 obtained within the time range.
 - If you click the green rectangle, you can view a more fine-grained log distribution. You can also view the query and analysis results on the Raw Logs tab.
- Raw Logs tab
 - On the Raw Logs tab, you can view the logs that match your search conditions.
 - Quick analysis: You can use this feature to analyze the distribution of a specific field within a specific period of time. For more information, see Quick analysis.
- Contextual query: If you click the 🗋 icon of a log and select **Context View**, you can view the context of the log. For more information, see Contextual query.

⑦ Note The contextual query feature supports only the log data that is collected by Logtail.

• LiveTail: If you click the 📄 icon of a log on the Raw Log tab, you can monitor logs in real time and extract important information. For more information, see LiveTail.

⑦ Note LiveTail can monitor and extract only the log data that is collected by Logtail.

- Log download: To download logs, click the download icon, select a method, and then click OK. For more information, see Download logs.
- Column settings: You can click the olicon and select Column Settings to specify the columns that you want to display in the table. The column names are field names, and the column content is used as field values.

?	Note				
То	view the log	content on	the tab,	select Content.	

- JSON configurations: You can click the interval is a specify the levels of JSON data.
- Tag configurations: On the Raw Data tab, you can click the 👩 icon and select Tag Configurations to hide fields that are less important.

2 Feb 23, 15:24:44	8	▶ >	©17	⊟iZb dtutZ
	v (content: {}		
		@timestamp:	"2022-02-23T07:2	24:43.026Z"
		@metadata:	{}	

Charts

- If you turn on Enable Analytics when you configure indexes for fields and use query statements to query logs, you can view the analysis results on the **Graph** tab.
- Log Service provides multiple chart types, such as tables, line charts, and bar charts. You can select a chart type to display analysis results. For more information, see Chart overview.
- Log Service allows you to create dashboards to perform real-time data analysis. You can click Add to New Dashboard to save query statements as charts to a dashboard. For more information, see Dashboard overview.
- Drill-down analysis allows you to view more details of analysis results. You can configure the drill-down parameters and add a chart to the dashboard. Then, you can click the values in the chart to view the analysis results in multiple dimensions. For more information, see Configure a drill-down event.
- LogReduce tab
- On the LogReduce tab, you can click Enable LogReduce to cluster similar logs. For more information, see LogReduce.
- Alerts

On the Search & Analysis page, click Save as Alert to create an alert monitoring rule for query results. For more information, see Alert overview.

Saved searches

On the Search & Analysis page, you can click Save Search to save a query statement as a saved search. For more information, see Saved search.

4.4.6. Download logs

Log Service allows you to download logs or query and analysis results to your on-premises machine. This topic describes how to download logs or query and analysis results.

Procedure

- 1. Log on to the Log Service console
- 2. In the Projects section, click the name of the project in which you want to download logs.
- 3. Click the picon next to the name of the Logstore whose logs you want to download and select Search & Analysis.

4. On the **Raw Logs** tab, click the icon.

- 5. In the Log Download dialog box, select a download method and complete the download as prompted.
 - Download Log in Current Page: downloads logs displayed on the current page as a file in the comma-separated values (CSV) format.
 - Download All Logs Using Command Line Tool: downloads all logs as prompted.

4.4.7. Enable Dedicated SQL

Compared with the Standard SQL feature, the Dedicated SQL feature has no limits on the number of concurrent operations or the amount of data that can be analyzed. You can use the Dedicated SQL feature to query a larger amount of data.

Prerequisites

- A Logstore is created. For more information, see Create a Logstore.
- Logs are collected. For more information, see Data collection overview.
- Indexes are configured. For more information, see Configure indexes.

Background information

If you use the Standard SQL feature to analyze a large amount of log data that is generated over a period of time, Log Service cannot scan all log data in a single query. To ensure timeliness, Log Service limits the amount of data that is scanned in each shard and returns some inaccurate results. In this case, we recommend that you increase the number of shards to increase computing resources. However, after you increase the number of shards, only new data that is written to the shards can be read for scanning. Historical data cannot be read for scanning. The number of consumers also increases.

To resolve this issue, Log Service provides the Dedicated SQL feature. The Dedicated SQL feature can efficiently analyze log data.

? Note

The Dedicated SQL feature and the Standard SQL feature are both available. You can choose between the features based on your business requirements.

Advantages

You can use the Dedicated SQL feature to analyze log data by using SQL statements. The Dedicated SQL feature has the following advantages over the Standard SQL feature:

- The Dedicated SQL feature can analyze hundreds of billions of data records with high performance.
- The Dedicated SQL feature allows up to 100 concurrent operations in each project. The Standard SQL feature allows only 15 concurrent operations.
- The Dedicated SQL feature is allocated exclusive resources. The performance of the Dedicated SQL feature is not affected by traffic bursts from other

Scenarios

lisers

The Dedicated SQL feature is suitable for the following scenarios:

- You need to analyze data with high performance. For example, you need to analyze data in real time.
- You need to analyze data that is generated over a long period of time. For example, you need to analyze data that is generated over a month.
- You need to analyze a large amount of data. For example, you need to analyze terabytes of data every day.
- You need to analyze data by using more than 15 concurrent SQL statements and display the analysis results based on multiple metrics from multiple dimensions.

Procedure

Log Service supports the following methods to enable the Dedicated SQL feature:

- Enable once: When you execute a query statement in a Logstore, click the 🔊 icon. The Dedicated SQL feature takes effect only on the query statements that you execute in the current Logstore.
- Enable permanently: If you turn on **Enable by Default**, the Dedicated SQL feature is enabled for the current project and takes effect on all query statements that you execute in the project, including the query statements for alerts and dashboards.

Enable once

- 1. Log on to the Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. On the Log Storage > Logstores tab, click the Logstore that you want to manage.
- 4. Click the conicon.

After you enable the Dedicated SQL feature, you can use the feature to query and analyze log data by using SQL statements. For more information, see Query and analyze logs.

Enable permanently

1. Log on to the Log Service console

2. In the Projects section, click the project that you want to manage.

3. Click the icon.

4. Move the pointer over the CUs of SQL-dedicated Instance parameter and click Modify.

5. In the Modify CUs of SQL-dedicated Instance panel, turn on Enable by Default and click OK.

4.4.8. Index data type

4.4.8.1. Overview

When you configure indexes, you can set the data type of a field to text, long, double, or JSON. This topic describes the index data types that are supported by Log Service.

Data types

The following table describes the supported data types.

Query type	Index data type	Description	Example
Basic query	text	The text type. You can use keywords and fuzzy matches to query logs.	<pre>uri:"login*" and method:"post"</pre>
	long	The numeric type. You can specify numeric ranges to query indexes of this type.	status in [200, 500]
	double	The floating-point type.	price>28.95
Combined query	JSON	Indicates that the index is a JSON field that supports nested queries. By default, the data type of the field is text. You can configure indexes of the text, long, and double types for the b elements at layer a in the a.b path format.	<pre>level0.key>29.95 and level0.key2:"action"</pre>
	text	Creates indexes for all fields in a log except the time field. The data type of the indexes is text.	error and "login fail"

4.4.8.2. Text type

This topic describes how to query text data.

Usage notes

Similar to search engines, Log Service queries text data based on terms. Therefore, you must set the Delimiter and Case Sensitive parameters when you configure indexes.

Case Sensitive switch

You can specify whether searches are case-sensitive. For example, you want to query a log entry that contains internalError .

- If you turn off Case Sensitive, searches are case-insensitive, and you can find the log entry by using the INTERNALERROR or Internalerror keyword.
- If you turn on Case Sensitive, searches are case-sensitive, and you can find the log entry only by using the internalError keyword.

Delimiter parameter

You can use delimiters to split the content of a log entry into multiple words. For example, you want to query a log entry that contains the following content:

/url/pic/abc.gif

- If you do not specify a delimiter, the entire string is processed as a single word in the /url/pic/abc.gif format. In this case, you can find the log entry by using the entire string as a keyword for exact match or by using the /url/pic/* keyword for fuzzy match.
- If you set the delimiter to a forward slash (/), the content is divided into the following three words: url, pic, and abc.gif. You can find the log entry by using one of the three words. You can also use part of each word to search for the log entry in fuzzy match mode.
 For example, you can find the log entry by using the url, abc.gif, or pi* keyword. You can also find the log entry by using the /url/pic/abc.gif keyword is split into the following search conditions: url and pic and abc.gif.

• If you set the delimiter to a forward slash (/) and a period (.), the content is split into the following four words: url , pic , abc , and gif .

? Note

You can specify appropriate delimiters to extend query ranges.

• Full Text Index switch

By default, after you turn on Full Text Index, the data type of all fields, except the time field, is set to text. You do not need to specify keys. For example, you want to query a log entry that consists of the following four fields:

```
time:2018-01-02 12:00:00
level:"error"
status:200
message:"some thing is error in this field"
```

[20180102 12:00:00],200,error,some thing is error in this field

? Note

- Prefixes are not required for full-text indexes. If you use error as a keyword, the level and message field values that contain error match the keyword.
- You must specify delimiters for full-text indexes. For example, if you specify a comma (,) as a delimiter, the status:200 string is processed as a single word. If you specify a colon (:) as a delimiter, the string is split into the following two words: status and 200
- Numbers are processed as text data. For example, you can find the log entry by using the keyword 200. The time field is not processed as text data.
- If the query statement is a key, for example, status , the log entry is matched.

4.4.8.3. Numeric type

When you configure indexes, you can set the data type of a field to a numeric type. Then, you can query the value of the field by value range.

Usage notes

- You can query the value of a field by using a numeric range only after you set the data type of the field to long or double.
- If the value of a log field is an integer, we recommend that you set the data type of the field to long when you configure indexes.

• If the value of a log field is a floating-point number, we recommend that you set the data type of the field to double when you configure indexes.

() Important

- If you set the data type of a field to long but the value of the field is a floating-point number, you cannot query the value of the field.
- If you set the data type of a field to long or double but the value of the field is a string, you cannot query the value of the field.
- If you set the data type of a field to long or double, you cannot use asterisks (*) or question marks (?) to query the value of the field in fuzzy match mode.
- If the value of a field is an invalid numeric value, you can query data by using the not key > -1000000 search statement. The not key > 1000000 search statement returns the log entries in which a field value is an invalid numeric value. -100000 can be replaced by a valid
 value that is less than or equal to the smallest valid value of the field in your log entries.

Sample search statements

Sample log entry

1 02-02 11:36:03	··· @17 78 1612236963 nginx_access_log
	tag:client_ip:47 166
	body_bytes_sent:2636
	client_ip:1 59
	host :www.mk.mock.com
	http_user_agent :Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.9 Safari/536.5
	region :cn-shanghai
	remote_addr:119 54
	remote_user:5xrtx
	request_length:1771
	request_method :GET
	request_time:34
	request_uri:/request/path-2/file-7
	status:200

Index configurations

Field Search					Automatic	Index Generation
	Enable Search					
Key Name	Туре	Alias	Case Sensitive	Delimiter: ?	Chinese	Enable Delete Analytics
body_bytes_sent	long \lor					
client_ip	text \lor			, ''';=()[]{}?@&<>/:\n\t\r	\bigcirc	
host	text \lor			, ''';=()[]{}?@&<>/:\n\t\r	\bigcirc	
http_user_agent	text \lor			, "";=()[]{}?@&<>/:\n\t\r	\bigcirc	
region	text \lor			, "";=()[]{}?@&<>/:\n\t\r	\bigcirc	
remote_addr	text \lor			, "";=()[]{}?@&<>/:\n\t\r	\bigcirc	
remote_user	text \checkmark			, ''';=()[]{}?@&<>/:\n\t\r	\bigcirc	
request_length	long 🗸					
request_method	text \lor			, "";=()[]{}?@&<>/:\n\t\r	\bigcirc	
request_time	long \lor					
request_uri	text \lor			, "";=()[]{}?@&<>/:\n\t\r	\bigcirc	
status	long \lor					

• Query statements

 $\circ\,$ To query the log entries in which the request duration is greater than 60 seconds, execute the following search statement:

```
request_time > 60
```

 To query the log entries in which the request duration is greater than or equal to 60 seconds and less than 200 seconds, execute one of the following search statements:

request_time in [60 200)
request_time >= 60 and request_time < 200</pre>

• To query the log entries in which the response status code is 200, execute the following search statement:

status = 200

4.4.8.4. JSON type

If the value of a field is in the JSON format, you can set the data type of the field to JSON when you configure indexes. This topic describes how to set the data type of a field to JSON and provides some examples.

Usage notes

- You can set the data type of a field in JSON objects to long, double, or text based on the field value, and turn on Enable Analytics to enable the
 analysis feature. After you turn on Enable Analytics, Log Service allows you to query and analyze fields in JSON objects.
- For partially valid JSON-formatted data, only the valid parts can be parsed in Log Service.
- The following example shows an incomplete JSON log entry. Log Service can parse the **conctent.remote_addr**, **content.request.request_length**, and **content.request_method** fields.

tent: {
remote_addr:"192.0.2.0"
request: {
request_length:"73"
request_method:"GE

() Important

- Log Service allows you to configure indexes for leaf nodes in JSON objects. However, you cannot configure indexes for child nodes that contain leaf nodes.
- You cannot configure indexes for fields whose values are JSON arrays or configure indexes for the fields in a JSON array.
- If the value of a field is of the Boolean type, you can set the data type of the field to text when you configure indexes.
- The format of a query statement in Log Service is Search statement | Analytic statement . In an analytic statement, you must enclose a field name by using double quotation marks ("") and enclose a string by using single quotation marks (").

Examples

The following table lists the keys included in the sample log entry. The data type of the message field is JSON.

Serial number Key Type

0	time	N/A
1	class	text
2	status	long
3	latency	double
4	message	json

Sample log entry:

<pre>0. time:2018-01-01 12:00:00 1. class:central-log 2. status:200 3. latency:68.75</pre>
4. message:
1
"methodName": "getProjectInfo",
"success": true,
"remoteAddress": "203.0.113.10:11111",
"usedTime": 48,
"param": {
"projectName": "ali-log-test-project",
"requestId": "d3f0c96a-51b0-4166-a850-f4175dde7323"
},
"result": {
"message": "successful",
"code": "200",
"data": {
"clusterRegion": "ap-southeast-1",
"ProjectName": "ali-log-test-project",
"CreateTime": "2017-06-08 20:22:41"
},
"success": true
}
}

The following figure shows an example on how to configure indexes.

Figure 1. Index configurations

Field Search							Automatic	Index Gene	ration
		Enable Search				Include	Frable		
	Key Name			Alias	Case Sensitive	Delimiter: ?	Chinese	Analytics	Delete
class		text	~			, ''';=()[]{}?@&<>/:\n\t\r	\bigcirc		$) \times$
info	info		\sim			, ''';=()[]{}?@&<>/:\n\t\r	\bigcirc		X
	methodName	text	\sim						$) \times$
	param.projectName	text	\sim						$) \times$
	param.requestId	text	\sim						$) \times$
	result.code	long	\sim						$) \times$
	result.message	text	\sim						$) \times$
	success	text	\sim						$) \times$
	usedTime	long	\sim						$) \times$
				+					
latency		long	\sim						$) \times$
status		long	\sim						$) \times$

The following settings are configured in the preceding figure:

• ① specifies that Log Service can query data of the string and Boolean types in JSON fields.

- ② specifies that Log Service can query data of the long type.
- ③ enables SQL analysis for specified fields.
- Query log data of the string and Boolean types.

? Note

You do not need to configure JSON fields.

• JSON maps and arrays are automatically expanded and can contain multiple layers. You must separate multiple layers with periods (.).

Query statement:

```
message.traceInfo.requestId : 92.137_1518139699935_5599
message.param.projectName : ali-log-test-project
message.success : true
message.result.data.ProjectStatus : Normal
```

• Query log data of the double and long types.

? Note

You must configure each JSON field. A JSON field cannot be contained in an array.

Query statement:

message.usedTime > 40

• Use SQL statements to analyze fields.

? Note

- You must configure each JSON field. A JSON field cannot be contained in an array.
- You must enclose a field name by using double quotation marks ("") or specify an alias for the field.

Query statement:

* | select avg("message.usedTime") as avg_time ,"message.methodName" group by "message.methodName"

4.4.9. Query syntax and functions

4.4.9.1. Search syntax

This topic describes how to use the search syntax that is provided by Log Service to specify search conditions. You can efficiently query logs based on the search syntax.

Search types

A search statement specifies one or more search conditions and returns the logs that meet the specified conditions. Searches are classified by indexing method into full-text searches and field-specific searches, or classified by precision into exact searches and fuzzy searches.

? Note

- If you configure both full-text indexes and field indexes, the configurations of the field indexes take precedence.
- Before you can specify a numeric range to query logs based on a field, you must set the data type of the field to double or long. If you do not set the data type of a field to double or long, or the syntax of the numeric range is invalid, Log Service performs a full-text search and the search result that is returned may be different from the expected result. For example, if you execute the <u>owner_id>100</u> search statement and the data type of the **owner_id** field is not double or long, logs that contain **owner_id**. > (non-delimiter), and **100** are returned.
- If you change the data type of a field from text to double or long, you can use only the equal sign (=) to query the logs that are collected before the change.

• Full-text searches and field-specific searches

Search type	Description	Example		
Full-text search	After you configure full-text indexes, Log Service splits a log into multiple words by using the delimiters that you specify. You can specify keywords and rules in a search statement to query logs. The keywords can be field names or field values.	PUT and cn-shanghai : returns the logs that contain the keywords PUT and cn-shanghai .		
Field-specific search	After you configure field indexes, you can query logs. To query logs, specify field names and field values in the key:value format. You can perform basic searches or combined searches based on the data types of the fields in the field indexes. For more information, see Data types.	<pre>request_time>60 and request_method:Ge* : returns the logs in which the value of the request_time field is greater than 60 and the value of the request_method field starts with Ge.</pre>		

• Exact searches and fuzzy searches

Search type	Description	Example
Exact search	Complete strings are used for queries.	 host:www.yl.mock.com : returns the logs in which the value of the host field is www.yl.mock.com. PUT : returns the logs that contain the keyword PUT.

Fuzzy search	You can add an asterisk (*) or a question mark (?) as a wildcard in the middle or at the end of a keyword to perform a fuzzy search. Each keyword must be 1 to 64 characters in length. If a keyword contains a wildcard, Log Service searches all logs and obtains up to 100 strings that match the keyword. Then, Log Service returns the logs that contain one or more of these strings. The more accurate a keyword is, the more accurate the search results are.	
	 Note A keyword cannot start with an asterisk (*) or a question mark (?). The long and double data types do not support asterisks (*) or question marks (?) in fuzzy searches. You can specify a numeric range when you perform a fuzzy search. Example: status in [200 299]. 	 addr* : searches for 100 strings that start with addr from logs, and returns the logs that contain one or more of these strings. host:www.yl* : searches for 100 strings that start with www.yl from the value of the host field. Then, Log Service returns the logs that contain one or more of these strings.
	A fuzzy search is performed based on samples by using the following mechanism:	g
	 If you enable the field indexing feature and specify a field to query logs, Log Service randomly obtains samples from the indexed data of the field and returns part of the search results. 	
	 If you enable the full-text indexing feature and do not specify a field to query logs, Log Service randomly obtains samples from the full-text indexed data ar returns part of the search results. 	

Operators

The following table describes the operators that are supported by search statements.

? Note

- The in operator is case-sensitive. Other operators are not case-sensitive.
- Log Service supports the following operators: sort, asc, desc, group by, avg, sum, min, max, and limit. If you want to use the preceding
 operators as keywords, you must enclose the operators in double quotation marks ("").
- The following list shows the priorities of the operators in descending order:
- i. Colons (:)
- ii. Double quotation marks ("")
- iii. Parentheses ()
- iv. and
- v. not
- vi. or

Operator	Description
and	The and operator. Example: request_method:GET and status:200 . If no syntax keyword exists among multiple keywords, the keywords are joined by using the and operator by default. For example, GET 200 cn-shanghai is equivalent to GET and 200 and cn-shanghai .
or	The or operator. Example: request_method:GET or status:200 .
not	The not operator. Examples: request_method:GET not status:200 and not status:200 .
()	This operator is used to increase the priority of the search conditions that are enclosed in parentheses (). Example: (request_method:GET or request_method:POST) and status:200 .
:	This operator is used for field-specific searches based on the key:value format. Example: <pre>request_method:GET</pre> . If a field name or field value contains reserved characters such as spaces and colons (:), enclose the field name or field value in double quotation marks (""). Example: "file info":apsara .
	This operator is used to enclose a syntax keyword. If a syntax keyword is enclosed in double quotation marks (""), the keyword is converted to an ordinary character. For example, "and" returns the logs that contain and. In this case, and is not an operator. In a field-specific search, the strings that are enclosed in double quotation marks ("") are considered as a whole string.
١	The escape character. This character is used to escape double quotation marks (""). Double quotation marks ("") can indicate themselves only after they are escaped. For example, if the content of a log is <pre>instance_id:nginx"01"</pre> , you can execute the <pre>instance_id:nginx\"01\"</pre> statement to search for the log.
	The wildcard character. This character is used to match zero, one, or multiple characters. Example: host:www.yl.mo*k.com .
*	ONOTE Log Service searches all logs and obtains up to 100 strings that meet the specified conditions. Then, Log Service returns the logs that contain one or more of the 100 strings and meet the search conditions.

?	The wildcard character. This character is used to match a single character. Example: host:www.yl.mo?k.com .
>	This operator is used to query the logs in which the value of a specified field is greater than a specified numeric value. Example: request_time>100 .
>=	This operator is used to query the logs in which the value of a specified field is greater than or equal to a specified numeric value. Example: request_time>=100 .
<	This operator is used to query the logs in which the value of a specified field is smaller than a specified numeric value. Example: request_time<100 .
<=	This operator is used to query the logs in which the value of a specified field is smaller than or equal to a specified numeric value. Example: request_time<=100 .
=	This operator is used to query the logs in which the value of a specified field is equal to a specified numeric value. Equal signs (=) and colons (:) have the same effect on fields of the double or long data type. For example, request_time=100 is equivalent to request_time:100.
in	This operator is used to query the logs in which the value of a specified field is within a specified numeric range. Brackets [] indicate a closed interval, and parentheses () indicate an open interval. A space character is used to separate two numbers in a numeric range. Examples: request_time in [100 200] and request_time in (100 200] . O Note The characters of in must be in lowercase.
source	 This operator is used to query the logs of a specified log source. Wildcard characters are supported. Example:
tag	This operator is used to query logs based on metadata. Example:tag_:receive_time:1609837139 .
topic	This operator is used to query the logs of a specified log topic. Example:topic:nginx_access_log .

Examples of search statements

Expected search result	Search statement
Logs that contain successful GET requests (status codes: 200 to 299)	request_method:GET and status in [200 299]
Logs that contain GET requests that are not sent from the China (Shanghai) region	request_method:GET not region:cn-shanghai
Logs that contain GET requests or POST requests	request_method:GET or request_method:POST
Logs that do not contain GET requests	not request_method:GET
Logs that contain successful GET requests or successful POST requests	(request_method:GET or request_method:FOST) and status in [200 299]
Logs that contain failed GET requests or failed POST requests	(request_method:GET or request_method:POST) not status in [200 299]
Logs that contain successful GET requests (status codes: 200 to 299) and in which the request duration is less than 60 seconds	request_method:GET and status in [200 299] not request_time>=60

Logs in which the request duration is equal to 60 seconds	<pre>request_time=60 </pre>						
Logs in which the request duration is greater than or equal to 60 seconds and is less than 200 seconds	<pre>request_time>=60 and request_time<200 request_time in (60 200)</pre>						
Logs in which the value of the http_user_agent field contains Firefox	http_user_agent:Firefox						
Logs in which the value of the http_user_agent field contains Linux and Chrome	<pre>http_user_agent:"Linux Chrome" http_user_agent:Linux and http_user_agent:Chrome</pre>						
Logs that contain and	"and" In this search statement, and is a common string but not an operator.						
Logs in which the value of the http_user_agent field contains Firefox or Chrome	http_user_agent:Firefox or http_user_agent:Chrome						
Logs in which the value of the file info field contains apsara	"file info":apsara						
Logs that contain strings that start with cn	cn*						
Logs in which the value of the region field starts with cn	region:cn*						
Logs in which the value of the $\ensuremath{\textit{region}}$ field contains $\ensuremath{\textit{cn}}\xspace^*$	region:"cn*"						
Logs in which the value of the region field ends with hai	Not supported.						
Logs that contain strings that start with mo, end with la, and contain one character between mo and la	mo?la						
Logs that contain strings that start with mo, end with la, and contain zero, one, or more characters between mo and la	mo*la						
Logs that contain strings that start with Moz and strings that start with Sa	Moz* and Sa*						
Logs whose topic is HTTPS or HTTP	topic_:HTTPS ortopic_:HTTP						
Logs that are collected from the 192.0.2.1 host	<pre>tag_:client_ip :192.0.2.1tag_:client_ip indicates the IP address of the host from which logs are collected.</pre>						
Logs in which the remote_user field is not empty	not remote_user:""						
Logs in which the remote_user field is empty	remote_user:""						
Logs that do not contain the remote_user field	not remote_user:*						

Logs in which the value of the remote_user field is null	not request_uri:"null"
Logs that contain the remote_user field	remote_user:*
Logs in which the value of the request_uri field is / request/path-2	request_uri:/request/path-2
Logs in which the value of the city field is not Shanghai	not city:Shanghai
Logs in which the value of the path field starts with /learn but does not contain/learn/level	path:/learn* not path:/learn/level

4.4.9.2. LiveTail

This topic describes how to use LiveTail to monitor and analyze logs.

Prerequisites

Logs are collected by Logtail. For more information, see Use Logtail to collect logs.

? Note

LiveTail can monitor and extract only the log data that is collected by Logtail.

Background information

In online O&M scenarios, you may need to monitor log data in real time and extract key information from the latest log data to identify causes of errors. If you use a traditional O&M method, you must run the **tail -f** command on each server to query log data. If you want to narrow the scope of the command output, you must run the **grep** or **grep -v** command to filter the log data by keyword. LiveTail that is provided in the Log Service console allows you to monitor and analyze online log data in real time. This helps you reduce your O&M workloads.

Benefits

- Logs are monitored in real time and filtered by keyword.
- Logs are collected and indexed based on the configurations of log collection.
- Log fields are delimited. This allows you to query contextual logs that contain delimiters.
- A log file can be found based on a log in the log file. This allows you to monitor the log file in real time without the need to log on to a server.

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. Click the icon next to the name of the Logstore in which logs are stored, and then select Search & Analysis.
- 4. On the **Raw Logs** tab, click the $rac{1}{|\mathbf{r}|}$ icon of a log.

5. In the LiveTail section, view logs.

After LiveTail is started, log data collected by Logtail is displayed on the page in real time. By default, the latest log data is displayed at the bottom of the list. You can view the latest log data without the need to scroll down. Up to 1,000 logs can be displayed on the page. If more than 1,000 logs are collected, the page is automatically refreshed to show the latest 1,000 logs.

6. If anomalies are detected in log data, click Stop.

After you stop LiveTail, logs in the log monitoring list are no longer updated. You can analyze and fix errors that are found when you monitor logs.

More operations

Operation	Description
Highlight strings	You can enter one or more strings in the Highlight field. The specified strings are highlighted in the LiveTail section.
Filter logs by string	You can enter one or more strings in the Filter By field. The LiveTail section displays only the logs that contain the specified strings.
Filter logs by field	You can select one or more fields from the Filter by Field drop-down list. The LiveTail section does not display the logs that contain the specified fields.
Stop LiveTail	You can click Stop to stop LiveTail. After you stop LiveTail, logs in the log monitoring list are no longer updated. You can analyze and fix errors that are found when you monitor logs.

4.4.9.3. LogReduce

This topic describes how to use the LogReduce feature of Log Service. You can enable the feature, view log clustering results and raw logs, and compare the number of clustered logs in different time periods.

Background information

The LogReduce feature allows you to cluster similar logs and extract patterns from the logs. The feature can cluster text logs in multiple formats. You can use the feature to perform O&M operations in DevOps scenarios. For example, you can use the feature to identify errors, detect anomalies, and roll back versions. You can also use the feature to detect intrusions in security scenarios. You can save log clustering results as charts to a dashboard and view the clustered data in real time.

Benefits

- You can cluster logs in multiple formats, such as Log4j logs, JSON-formatted logs, and single-line logs.
- · You can cluster hundreds of millions of logs within seconds.
- You can cluster logs in a variety of modes.
- You can retrieve raw logs based on pattern signatures.
- You can compare patterns that are extracted in different time periods.
- You can adjust the precision of log clustering based on your business requirements.

Enable the LogReduce feature

By default, the LogReduce feature is disabled.

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. Click the icon next to the name of the Logstore that you want to manage, and then select Search & Analysis.

4. Enable the LogReduce feature.

- i. Choose Index Attributes > Attributes.
- If the indexing feature is not enabled, click Enable.
- ii. In the Search & Analysis panel, turn on LogReduce.

⑦ Note

If a Logstore is a dedicated Logstore for a cloud service or an internal Logstore, you should turn off **Auto Update**, then you can turn on **LogReduce**.

iii. Optional:Configure an allowlist or denylist to filter fields.

You can filter logs based on keywords. Logs that are filtered based on keywords are automatically clustered.

- iv. Click OK
- v. In the dialog box of Search & Analysis Settings, click OK.

? Note

Modifications (such as changing the delimiter, enabling statistics, and enabling case-sensitivity) only take effect for new data. You can modify only the dedicated Logstores of some special cloud services. If you delete indexes, you cannot use the report or alerting feature.

View raw logs and the log clustering results

1. On the Search & Analysis page, enter a search statement in the search box, specify a time range, and then click Search & Analyze.

You can use only search statements to filter logs. You cannot use analytic statements to filter logs because the LogReduce feature cannot cluster analysis results.

2. Click the LogReduce tab to view the log clustering results.

On the LogReduce tab, you can view the filtered log clustering results.

Parameter	Description
Number	The ordinal number of the log cluster.
Count	The number of logs for the pattern in the specified query time range.
Pattern	The log pattern. Each log cluster has one or more sub-patterns.

• Move the pointer over a number in the **Count** column to view the sub-patterns of the log cluster. You can also view the percentage of each subpattern in the log cluster. Click the plus sign (+) next to a number in the Count column to expand the sub-pattern list.

• Click a number in the Count column. You are redirected to the Raw Logs tab. On this tab, you can view the raw logs of the pattern.

Change the precision of log clustering

On the LogReduce tab, you can adjust the Pattern Count slider to change the precision of log clustering.

- If you adjust the slider toward Many, you can obtain a more precise log clustering result that has more detailed patterns.
- If you adjust the slider toward Little, you can obtain a less precise log clustering result that has less detailed patterns.

Compare the number of logs that are clustered in different time periods

[?] Note

1. On the LogReduce tab, click Log Compare.

2. Specify a time range and click **OK**.

For example, if you set the time range to 15 minutes when you query logs and specify **1Day** for **Log Compare**, the start time and end time of log comparison are automatically displayed. The time ranges for comparison are the previous 15 minutes and the 15 minutes on the previous day.

Number	Pre_Count	🗘 Count	Diff	Pattern	Copy Query	Log Compare	✓ Add to New Dashboard
1	203	<u>7,890</u>	+7,687	kind:Event apl/Version.audit.k3s.iol/1beta1 metadata.{"creationTimestamp":"2019-0 6 requestURI:/*/** k8s.io/******	4-11TC 5Minutes 4Hours	15Minutes 1Hour 1Day 1Week	litID: * stage:ResponseComplete
2	2,841	<u>2,955</u>	+114 7 4.019	kind: Event ap/Version: audit k8s.iolv1beta1 metadata: ("creationTimestamp" "2019-0 stage: ResponseComplete requestURI/Japis "*** timeout=32s verb get user: ("userna responseStatus: ("metadata" (), "code":200) requestReceivedTimestamp:2019-04-11 ("authorization.k8s.ioldecision", "allow", "authorization.k8s.iolreason"."")	4-11TC 30Days me":"s T02: * Start Time:	2019-04-10 10:17:15	* : * auditID: * ;"]} sourceIPs:["127.0.0.1"] tations:
3	0	<u>2,289</u>	+2,289	kind: Event ap/Version: audit.k8s.io/v1beta1 metadata.["creationTimestamp"":2019-0 requestURI/api/v1/namespaces/* / * / ****** verb get user:["username"":system: ** ","namespace:" * ","name"** ** ap/Version"."v1") responseStatus.["metadata"; 11T02: * , * , * annotations.["authorization.k8s.io/decision"."aillow","authorization.k8	End Time: 4-11TC ," * code . } requestr s.io/reason":""}	2019-04-10 10:32:15 OK Received minestamp:2019-04-11	iitiD: * stage:ResponseComplete * "] objectRef:["resource": * roz: * : * . * stageTimestamp:2019-04-
4	0	<u>1,801</u>	+1,801	kind:Event apiVersion:audit.k8s.ioV1beta1 metadata;["creationTimestamp","2019-0 requestURI:Japis/ *_k8s.io/ * / ******* authorization.k8s.io/ *	4-11T02: * : * "} level: '	* timestamp:2019-04-11T02: * :	* auditID: * stage:ResponseComplete
-							

Parameter	Description
Number	The ordinal number of the log cluster.
Pre_Count	The number of logs for the pattern in the time range that is specified by Log Compare.
Count	The number of logs for the pattern in the time range that is specified for the query.
Diff	The difference between the number of logs in the Pre_Count column and the number of logs in the Count column, and the growth rate.
Pattern	The log pattern.

Examples of query statements

- You can use query statements to obtain log clustering results.
- Obtain log clustering results.
- Query statement

* | select a.pattern, a.count,a.signature, a.origin_signatures from (select log_reduce(3) as a from log) limit 1000

? Note

When you view log clustering results, you can click Copy Query to obtain the query statement of the log clustering results.

• Parameter settings

Modify the parameter settings in **log_reduce(precision)** of the query statement. The precision parameter specifies the precision of log clustering. A smaller value indicates a higher precision and more patterns. Valid values: 1 to 16. Default value: 3.

• Returned fields

You can view log clustering details on the Graph tab.

Parameter	Description
pattern	The log pattern.
count	The number of logs for the pattern in the time range that is specified for the query.
signature	The signature of the log pattern.
origin_signatures	The secondary signature of the log pattern. You can use the secondary signature to retrieve the raw logs.

• Compare the number of logs that are clustered in different time periods.

Query statement

* | select v.pattern, v.signature, v.count, v.count_compare, v.diff from (select compare_log_reduce(3, 86400) as v from log) order by v.di ff desc limit 1000

? Note

When you use **Log Compare** to compare log clustering results in different time periods, you can click **Copy Query** to obtain the query statement of the log clustering results.

- Parameter settings
- Modify the parameter settings in compare_log_reduce(precision, compare_interval) of the query statement.
- The precision parameter specifies the precision of log clustering. A smaller value indicates a higher precision and more patterns. Valid values: 1 to 16. Default value: 3.
- The compare_interval parameter specifies the time difference between the two time ranges for comparison. The value is a positive integer. Unit: seconds.
- Returned fields

Parameter	Description
pattern	The log pattern.
count_compare	The number of logs for the pattern in the previous time range that is specified for comparison.
count	The number of logs for the pattern in the time range that is specified for the query.
diff	The difference between the numbers of logs in the count and count_compare columns.
signature	The signature of the log pattern.

Disable the LogReduce feature

If you no longer need to use the LogReduce feature, you can disable the feature.

- 1. On the Search and Analysis page of the Logstore for which you want to disable this feature, choose Index Attributes > Attributes.
- 2. Turn off LogReduce.
- 3. Click **OK**.

4.4.9.4. Contextual query

This topic describes the contextual query feature provided in the Log Service console. You can use this feature to query the full context of the log file from which specified logs are obtained.

Prerequisites

• Logs are collected by Logtail. For more information, see Use Logtail to collect logs.

- ⑦ Note The contextual query feature is supported only for log data that is collected by Logtail.
- The indexing feature is enabled and indexes are configured. For more information, see Configure indexes.

Background information

To perform a contextual query, you must specify a source server, a source file, and a log whose context you want to query. You can obtain the logs that precede or follow the specified logcollected from the log file of the server. This helps you identify and resolve errors.

Scenario

For example, a transaction on an online-to-offline (O2O) takeout website is recorded in an application log file on a server. You must perform the following steps to complete a transaction: logon to the website, browse products, select a product, add the product to the shopping cart, place an order, pay for the order, deduct the order amount, and generate the order.

If the order fails, the O&M engineers must identify the cause of the failure at the earliest opportunity. In traditional contextual queries, the O&M engineers must be authorized by an administrator before they can log on to each server on which the O2O application is deployed. After the authorization is complete, the O&M engineers can search application logs files by order ID to identify the cause of the failure.

In Log Service, the O&M engineers can perform the following steps to locate the cause of the failure:

- 1. Install Logtail on the server. Create a machine group and a Logtail configuration in the Log Service console. Then, enable Logtail to upload incremental logs to Log Service.
- 2. On the Search & Analysis page of the Log Service console, specify a time range and find the log that records the failure based on the order ID.
- 3. After you find the log, scroll up until other related logs are found, for example, a log that records a credit card payment failure.



? Note

The contextual query feature does not support syslog logs.

Benefits

- You can identify the causes of failures without the need to modify applications or log file formats.
- You can query the context of a log from a log file that is collected from a server in the Log Service console. You do not need to log on to the server to query the context.
- You can specify a time range to find suspicious logs before you perform a contextual query in the Log Service console. This improves troubleshooting efficiency.
- You do not need to worry about data loss that is caused by insufficient server storage or log file rotation. You can view historical log data in the Log Service console at any time.

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project in which you want to perform contextual queries.
- 3. Click the price on next to the name of the Logstore in which you want to perform contextual queries, and then select Search & Analysis.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.
- 5. On the **Raw Logs** tab, find the log whose context you want to query and click the icon.

6. On the page that appears, scroll up and down to view the contextual logs.

- To scroll up, click Old.
- To scroll down, click New.
- To highlight specific strings, enter the strings in the Highlight field. The specified strings are highlighted in the Context View section.
- To filter logs by string, enter strings in the Filter By field. The Context View section displays only the logs that contain the specified strings.

Context View

< Old	Selected Log	New >	All Fields	~	Enter		Filter	request \times		Highlight												
tag_:_	hostname_:iZj6cg59xy	2cbm4ulet3zZ	_tag_:_path_;	/var/log	g/kubernetes/kubernetes-c4dee76	6e927ca4ca	a58372ebd	f6b02a25e.audit														
-29 [Fi a o o s u a u	<pre>eb 5, 13:41:36]tag mnotations: {"auths piversion: audit.ks bjectRef: {"resource equestReceivedTimes tageTimestamp: 2024 ser: {"username":"s p=mcnager#CeECt+0 serAgent: yurt-mana</pre>	_:_hostname_ s.io/v1 audit e":"leases","r tamp: 2024-02- -02-05T05:41:3 ystem:3=nvfor % diwszb"],"au ger/v0.0.0 (11	: 12j6cg59xy2ct midar is inc":"al ID: 02546546 is 05705:41:36.597 16.6025172 «raunt twine sys thentication.ku	-syste 606Z i bernet	<pre>et322tag :_ path_: /va 'stimulations'stat.informer 'states'state</pre>	ar/100000 Event len ger", "uid De Shoo 45 cied as. verb: upd	vel: Neta ""959fa	<pre>h c)acter2c)abi data 624-6636-46d5-bfm 624-6636-46d5-bfm 55 600020:f74"]}}</pre>	a-0543baedb25e Stem/leases/clou groups":["system	apiGroup d-yurt-new serviceacc	<pre></pre>	ndimp)" of () diom.kur.to", seStatus: ("r ten:senvignar	ikotaraci ","apivers: "metadata" argountsaku	<pre>ion":"v1" :{},"code when system</pre>	manager-ro ,"resource ":200} so m","system	le\' to s Version": µrceIPs: ∷authenti	:rviceAcco "J.#200#") ["" " " " " cated"],"e	iunt \"yu } e.77*1] extra":{*	urt-manager stage: Res "authentica	r/kube-syster sponseComple1 ation.kuberne	n\""} ∷e stes.io/pod	1-nane" : ["yu

4.4.9.5. Saved search

If you need to frequently view the result of a query statement, you can save the query statement as a saved search. Log Service provides the saved search feature to save the required data query and analysis operations. You can use a saved search to quickly perform query and analysis operations.

Prerequisites

The indexing feature is enabled and indexes are configured. For more information, see Configure indexes.

Background information

If you need to frequently view the result of a query statement, you can save the query statement as a saved search. Then, you can click the name of the saved search on the left side of the Search & Analysis page to execute the query statement and view the result.

You can also use the saved search in alert rules. Log Service periodically executes the query statement of the saved search and sends alert notifications if the query result meets the preset condition.

Create a saved search

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. Click the micron next to the name of the Logstore that you want to manage, and then select Search & Analysis.

4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.

A query statement consists of a search statement and an analytic statement in the **Search statement|Analytic statement** format. For more information, see Search syntax and SQL syntax and functions.

- 5. In the upper-right corner of the page, click **Save Search**.
- 6. In the Saved Search Details panel, configure the parameters. The following table describes the parameters.

Saved Search De	tails	×
* Saved Search Name	stage	
Attributes		
Logstores	audit-c03ff60740131455e931115et	83832ea8
Topic	The query statement of the current	query. It is empty if you do not set a t
Query	* SELECT stage, COUNT(*) as number	GROUP BY stage LIMIT 10
	Select the query statement to generate a pl down configuration to replace the variable.	aceholder variable. You can configure a drill-
Variable Config		
Variable Name:	Default Value:	Matching Mode:
stage	stage	Global Match V
Result		
* SELECT \${stage} . C	COUNT(*) as number GROUP BY \${stage} LI	VIT 10
Parameter		Description
Saved Search Na	ime	The name of the saved search. The name must be 3 to 63 characters in length.
		Select the required content of the query statement in the Query field and click Generate Variable to generate a placeholder variable.
		• Variable Name: the name of the placeholder variable.
		• Default Value : the content that you select from the Query field.
		 Matching Mode: the match mode. You can use the match mode to replace the default value by triggering a drill-down event. Valid values: Global Match and Exact Match.
Variable Config		For example, you set Event Action to Open Saved Search for a chart when you configure drill-down analysis for the chart, and specify the saved search. The Variable of the chart is the same as the Variable Name of the saved search. When you click the chart value, you are redirected to the saved search. The Default Value of the placeholder variable is replaced by the chart value that triggers the drill-down event. Then, the new query statement is executed. For more information, see Configure a drill-down event.
		⑦ Note Before you can set Event Action to Open Saved Search, you must create a saved search and configure variables.

7. Click **OK**.

After you create a saved search, click the vicen next to the search box on the Search & Analysis page of the Logstore, and click the name of the saved search to quickly perform query and analysis operations.

Modify a saved search

- 1. In the left-side navigation pane, choose **Resources > Saved Search**.
- 2. In the Saved Search list, click the saved search that you want to modify.
- 3. Enter a query statement and click Search & Analyze.

A query statement consists of a search statement and an analytic statement in the **Search statement**|**Analytic statement** format. For more information, see Search syntax and SQL syntax and functions.

- 4. In the upper-right corner of the page, click Modify Saved Search.
- 5. In the **Saved Search Details** panel, modify the settings and click **OK**.

Obtain the ID of a saved search

After you create a saved search, you can use the ID of the saved search to embed the saved search page to a self-managed web page.

1. In the left-side navigation pane, choose **Resources > Saved Search**.

- 2. In the Saved Search list, click the saved search whose ID you want to obtain.
- 3. Obtain the ID of the saved search in the URL.

4.4.9.6. Quick analysis

Log Service provides the quick analysis feature that allows you to analyze the distribution of a field within a specified time range in an efficient manner.

Prerequisites

Indexes are configured and the analysis feature is enabled for specified fields. For more information, see Configure indexes.

Features

• Allows you to analyze the first 100,000 log entries that are returned for a query.

? Note

When you perform quick analysis on log entries within a specified time range, Log Service collects the first 100,000 log entries. If you use a saved search to query all data in a Logstore, you must delete the Limit 100000 clause.

- Groups fields of the text type and provides statistics about the top 10 groups.
- Generates **approx_distinct** statements for fields of the text type.
- Supports histogram-based statistics about the approximate distribution of fields of the long and double type. Histogram-based statistics group sampling data and calculate the average value of each group.
- Searches for the maximum, minimum, average, or sum of fields of the long and double type.
- Generates a query statement based on quick analysis.

Procedure

- 1. Log on to the Log Service console
- 2. In the Projects section, click the project in which you want to perform quick analysis.
- 3. Click the name of the Logstore in which you want to perform quick analysis.
- 4. On the Raw Logs tab, click the field that you want to analyze in the Quick Analysis column.

Raw Logs Grap	h	LogReduce	
Quick Analysis			> Go to Page View
Search by field	Q	1 Mar 31, 17:32:48 🗒 … @ 11 nginx_access_log	A
body_bytes_sent	-	body_bytes_sent:1293	
		client_ip:	
client_ip	•	host: WWW.Z	
host	-	http_host:m	
	http_user_agent :Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:14.0) Gecko/20100101 Firefox/14.0.1		
nttp_user_agent	•	http_x_forwarded_for:14	

- Provide grouping statistics for fields of the TEXT type and approximate distribution histogram-based statistics for fields of the LONG and DOUBLE type. For more information, see Text type or Long and double types.
- Provide query statements.

Click the cincon next to the destination field. You are redirected to the **Graph** tab. A query statement for grouping statistics is provided in the search box.

• Calculate the number of unique values of a field.

In the Quick Analysis column, click **Count Distinct Values** under the destination field. You can obtain the number of unique values of the ${\rm Quick}_{\rm ReyName}$ field.

Display field names or aliases

Click the 🙀 icon to specify whether to display field names or aliases. Aliases can be set when you configure indexes. For example, if you set the alias of **host_name** to **host_host** is displayed in the Quick Analysis column after you select **Show Field Aliases**.

⑦ Note
If you do not set an alias for a field, the field name is displayed after you select Show Field Aliases .

Text type

The quick analysis feature provides grouping statistics for fields of the TEXT type. If you use this method, the first 100,000 log entries are grouped and the ratios of the top 10 groups are returned. For example, you can obtain the following result based on grouping statistics of **request_method**. The GET method is the most common request method.

Quick Analysis		
request_method GET		
POST	54.25%	
PUT	37.26%	
DELETE	4.62%	
	3.87%	
approx_distinct		

Long and double types

• Display approximate distribution by using histograms.

The number of field values of the LONG and DOUBLE types is large. The preceding grouping analytics method is not suitable for the LONG or DOUBLE type. Simple Log Service assigns field values into 10 buckets and displays the approximate distribution of the values in a histogram. The following figure shows the approximate distribution of the **request_time** field. This distribution of field value indicates that most of the request periods are distributed around 1.346 ms.

Quick Analysis			
request_method			
request_time			
0.05624418604651	162		
	12.82%		
0.17316			
	11.18%		
0.26931746031746	603		
	9.39%		
0.38196774193548	3383		
	9.24%		
0.47996721311475	415		
	9.09%		
0.5410303030303030	031		
	4.92%		
0.61433846153846	616		
	9.69%		
0.71683561643835	561		
	10.88%		
0.81859322033898	331		
	8.79%		
0.937			
	14.01%		
Max Min Avg Sum			

Quick analysis on the maximum value, minimum value, average and sum of fields.
 You can click Max under a field to search for the maximum value, Min to search for the minimum value, Avg to calculate the average value, and Sum to calculate the sum of fields.

4.4.10. SQL syntax and functions

4.4.10.1. General aggregate functions

An aggregate function is used to calculate a set of values and return a single value. This topic describes the syntax of aggregate functions. This topic also provides examples on how to use the functions.

Function	Description	Example
arbitrary(KEY)	Returns a random, non-null value from a specified column.	The following query statement returns an arbitrary value from the request_method column: * SELECT arbitrary(request_method) AS request_method
avg(KEY)	Calculates the arithmetic mean of the values in a specified column.	The following query statement returns the projects whose average latency is greater than 1,000 microseconds. You can execute the statement to analyze the write latency of the projects. method: PostLogstoreLogs SELECT avg(latency) AS avg_latency, Project GROUP BY Project HAVING avg_latency > 1000
checksum(KEY)	Calculates the checksum of a specified column and returns a result that is encoded in Base64.	The following query statement calculates the checksum of the request_method column: * SELECT checksum(request_method) The returned result is D2UmTL3octI= .
count(*)	Calculates the number of log entries.	The following query statement calculates the number of page views (PVs): * SELECT count(*) AS PV
count(KEY)	Calculates the number of the log entries that contain a specified field. If the field value of a log entry is null, the log entry is not counted.	The following query statement calculates the number of the log entries that contain the request_method field: * SELECT count(request_method)
count(1)	Calculates the number of log entries. This function is equivalent to $\operatorname{count}(*)$.	The following query statement calculates the number of PVs: * SELECT count(1) AS PV

Cloud Defined Storage

count_if(KEY)	Calculates the number of log entries that meet a specified condition.	The following query statement calculates the number of requests for the value of the url field. The value ends with abc . * SELECT count_if(url like '%abc')
geometric_mean(KEY)	Calculates the geometric mean of the values in a specified column.	The following query statement calculates the geometric mean of request durations: * SELECT geometric_mean(request_time)
max_by(KEY_01,KEY_02)	Returns the value of KEY_01 that is associated with KEY_02 whose value is the maximum value.	The following query statement returns the point in time when the highest consumption occurs: * SELECT max_by(UsageEndTime, PretaxAmount) as time
max_by(KEY_01,KEY_02,n)	Returns the n values of KEY_01 that is associated with KEY_02 whose values are the first n maximum values. The returned result is a JSON array.	The following query statement returns the three request methods that have the longest request durations: * SELECT max_by(request_method,request_time,3) The returned result is ["GET","PUT","DELETE"] .
min_by(KEY_01,KEY_02)	Returns the value of KEY_01 that is associated with KEY_02 whose value is the minimum value.	The following query statement returns the request method whose request duration is the shortest: * SELECT min_by(request_method,request_time)
min_by(KEY_01,KEY_02,n)	Returns the n values of KEY_01 that is associated with KEY_02 whose values are the first n minimum values. The returned result is a JSON array.	The following query statement returns the three request methods that have the shortest request durations: * SELECT min_by(request_method,request_time,3) The returned result is ["GET","PUT","DELETE"] .
max(KEY)	Queries the maximum value of a specified column.	The following query statement queries the longest request duration: * SELECT max(request_time)
min(KEY)	Queries the minimum value of a specified column.	The following query statement queries the shortest request duration: * SELECT min(request_time)
sum(KEY)	Calculates the total value of a specified column.	The following statement calculates the total size of daily NGINX traffic: * select date_trunc('day',time) AS time, sum(body_bytes_sent) AS body_bytes_sent GROUP BY time ORDER BY time
bitwise_and_agg(KEY)	Returns the result of the bitwise AND operation for the values of a specified column. The returned result is in the two's complement format.	The following query statement performs a bitwise AND operation on all values of the request_time column: * SELECT bitwise_and_agg(request_time)
bitwise_or_agg(KEY)	Returns the result of the bitwise OR operation for values of a specified column. The returned result is in the two's complement format.	The following query statement performs a bitwise OR operation on all values of the request_time column: * SELECT bitwise_or_agg(request_time)

4.4.10.2. Security check functions

Security check functions in Log Services are designed based on the globally shared WhiteHat Security asset library. This topic describes security check functions that you can use to check whether an IP address, domain name, or URL in logs is secure.

Scenarios

• O&M personnel of enterprises and institutions in Internet, gaming, information, and other industries that require robust O&M services can use security check functions to identify suspicious requests or attacks. They can also use the functions to implement in-depth analysis and defend against potential attacks.
O&M personnel of enterprises and institutions in banking, securities, e-commerce, and other industries that require strong protection for internal
assets can use security check functions to identify requests to suspicious websites and downloads initiated by trojans. Then the O&M personnel can
take immediate actions to prevent potential losses.

Features

- Reliability: built upon the globally shared WhiteHat Security asset library that is updated in a timely manner.
- · Efficiency: capable of screening millions of IP addresses, domain names, and URLs within seconds.
- Ease of use: supports the analysis of network logs by using the security_check_ip, security_check_domain, and security_check_url functions.
- Flexibility: supports interactive queries, report creation, and alert configurations and subsequent actions.

Functions

Function	Description	Example
security_check_ip	Checks whether an IP address is secure.The value 1 indicates that the specified IP address is suspicious.The value 0 indicates that the specified IP address is secure.	<pre>select security_check_ip(real_client_ip)</pre>
security_check_domain	Checks whether a domain name is secure.The value 1 indicates that the specified domain name is suspicious.The value 0 indicates that the specified domain name is secure.	<pre>select security_check_domain(site)</pre>
security_check_url	Checks whether a URL is secure.The value 1 indicates that the specified URL is suspicious.The value 0 indicates that the specified URL is secure.	<pre>select security_check_domain(concat(host, url))</pre>

Examples

Check external suspicious requests and generate reports

For example, an e-commerce enterprise collects logs from its NGINX servers and wants to scan suspicious client IP addresses. To do this, the enterprise can pass the ClientIP field in logs that are collected from the NGINX servers to the security_check_ip function and filter out IP addresses associated with the returned value 1. Then the enterprise can query the countries where the IP addresses are located and ISPs to which the IP addresses belong.

SQL statement for this scenario:

* | select ClientIP, ip_to_country(ClientIP) as country, ip_to_provider(ClientIP) as provider, count(1) as PV where security_check_ip(ClientIP) = 1 group by ClientIP order by PV desc

Display the ISPs and countries in a map.

client_ip	\$ Q.	country 🗘 🗘	provider 🗘 🤤	PV \$ 0.
180	3	CN	E	3
103		CN		3
180	7	CN	E j	1

Check internal suspicious requests and send alerts

For example, a securities operator collects logs of its internal devices that access the Internet through gateways. To check requests to suspicious websites, the operator can run the following statement:

* | select client_ip, count(1) as PV where security_check_ip(remote_addr) = 1 or security_check_site(site) = 1 or security_check_url(concat(site, url)) = 1 group by client_ip order by PV desc

The operator can save this statement as a saved search and configure an alert. An alert is triggered when a client frequently accesses suspicious websites. The statement in the alert can be configured to run every five minutes to check if a client has frequently (more than five times) accessed suspicious websites in the past one hour. The following figure shows the configurations of an alert.

Create Alert					×
Alert	Configuration			Notifications	
* Alert Name	alarm				5/64
* Add to New Ø Dashboard	Create \lor	access_	alarm		12/64
* Chart Name	alarm				5/64
Query * Search Period	* select client_ip, or security_check group by client_ip	, count(1) as _site(site) = ` order by PV lative) ▼	PV where securi I or security_che desc	ty_check_ip(remote_ ck_url(concat(site, ur	addr) = 1 1)) = 1
* Check Frequency	Fixed Interval	√ 15		+ Minutes	\sim
* Trigger Condition 🕜	pv>5				4/128
Advanced >	Five basic operators (/), and modulo (%). greater than or equa (==), not equal to (I= (I~).Documentation	are support Eight compa I to (>=), less ;), regex mat	ed: plus (+), minu rrison operators a s than (<), less th ch (=~), and neg	us (-), multiplication (* are supported: greate lan or equal to (<=), e ated regex match), division r than (>), equal to

4.4.10.3. Map functions and operators

This topic describes the syntax of map functions and operators. This topic also provides examples on how to use the functions and operators. The following table describes the map functions and operators that are supported by Log Service.

() Important

If you want to use strings in analytic statements, you must enclose the strings in single quotation marks ("). Strings that are not enclosed or are enclosed in double quotation marks ("") indicate field names or column names. For example, 'status' indicates the status string, and status or "status" indicates the status log field.

Function Syntax		Description
Subscript operator	[X]	Returns the value of a key from a map.
cardinality function	cardinality(<i>x</i>)	Returns the size of a map.
element_at function	element_at(<i>x</i> , <i>key</i>)	Returns the value of a key from a map.
histogram function	histogram(<i>x</i>)	Groups query and analysis results and returns data in the JSON format.
histogram_u function	histogram_u(<i>x</i>)	Groups query and analysis results and returns data in multiple rows and multiple columns.
mon function	map()	Returns an empty map.
map runction	map(<i>x</i> , <i>y</i>)	Returns a map that is created by using two arrays.
map_agg function	map_agg(<i>x</i> , <i>y</i>)	Returns a map that is created by using x and y , x is a key in the map. y is the value of the key in the map. If y has multiple values, a random value is extracted as the value of the key.
map_concat function	map_concat(x, y)	Returns the union of multiple maps.

map_filter function	map_filter(<i>x</i> , <i>lambda_expression</i>)	Filters elements in a map based on a lambda expression.
map_keys function	map_keys(<i>x</i>)	Returns an array that consists of all keys in a map.
map_values function	map_values(x)	Returns an array that consists of all values in a map.
multimap_agg function	multimap_agg(<i>x, y</i>)	Returns a multimap that is created by using x and y . x is a key in the multimap. y is the value of the key in the multimap. The value is of the array type. If y has multiple values, all the values are extracted as the values of the key.

Subscript operator

The subscript operator is used to return the value of a key from a map.

- Syntax
- [x]
- Parameters

Parameter	Description
x	The value of this parameter is of the varchar type.

- Return value type
 - An arbitrary data type.
- Example

In a log that is transformed by a data transformation job, the value of the **etl_context** field is of the map type. You can use the subscript operator to obtain the value of the **project** key from the value of the **etl_context** field.

• Sample field

	<pre>etl_context: { project:"datalab-148****6461-cn-chengdu" logstore:"internal-etl-log" consumer_group:"etl-83****4d1965" consumer:"etl-b2d40ed****c8d6-291294" shard_id:"0" }</pre>			
0	Query statement			
	* SELECT try_cast(json_parse(etl_context) #	AS map(varchar, varchar))['project']		
٥	Query and analysis result			
	_col0	\$Q.		
	datalab-14 3461-cn-chengdu			
cai	dinality function			
The	cardinality function is used to return the size of	a map.		
• Sy	intax			
0	cardinality(x)			
• Pa	rameters			
	Parameter	Description		
	x	The value of this parameter is of the map type.		
• Re	eturn value type			
Tł	e bigint type.			
• E>	ample			
Us re	e the histogram function to obtain the number of quest methods.	of requests for each request method. Then, use the cardinality function to obtain the number of		
٥	Query statement			
<pre>* SELECT histogram(request_method) AS request_method, cardinality(histogram(request_method)) AS "kinds"</pre>				

• Query and analysis result

request_method	\$ Q	kinds	\$Q
{"DELETE":5, "POST":7, "GET":41, "PUT":4}		4	

element_at function

The element_at function is used to return the value of a key from a map.

- Syntax
- element_at(x, key)

• Parameters

Parameter	Description
x	The value of this parameter is of the map type.
key	The value of this parameter is a key in the specified map.

Return value type

An arbitrary data type.

• Example

Use the histogram function to obtain the number of requests for each request method. Then, use the element_at function to obtain the value of the **DELETE** field.

0	Query	statement
---	-------	-----------

*
SELECT
histogram(request_method) AS request_method,
<pre>element_at(histogram(request_method),'DELETE') AS "count"</pre>

0	Query	and	ana	lysis	result
---	-------	-----	-----	-------	--------

request_method	\$ Q	count	\$ Q,
{"HEAD":9,"DELETE":140,"POST":319,"GET":1298,"PUT":337}		140	

histogram function

The histogram function is used to group query and analysis results and return data in the JSON format. This function is equivalent to * | SELECT count (*) GROUP BY x .

- Syntax
- histogram(x)
- Parameters

	Parameter	Description
	x	The value of this parameter is of an arbitrary data type.
•	Return value type	

The map type.

- Example
 - Use the histogram function to obtain the number of requests for each request method.
- Query statement

*	I.	SELECT	histogram(r	equest_	method)	AS	request	method	
---	----	--------	-------------	---------	---------	----	---------	--------	--

0	Query	and	analysis	result
---	-------	-----	----------	--------

request_method	\$ Q.
{"HEAD":30, "DELETE":564, "POST":1382, "GET":5420, "PUT":1334}	

histogram_u function

The histogram_u function is used to group query and analysis results and return data in multiple rows and multiple columns.

- Syntax
- histogram_u(x)

• Parameters

Parameter	Description
x	The value of this parameter is of an arbitrary data type.

- Return value type
- The bigint type.
- Example

Use the histogram_u function to obtain the number of requests for each request method and then display the number on a column chart.

• Query statement

*|SELECT histogram_u(request_method) as request_method

Query and analysis result



map function

The map function is used to return an empty map or return a map that is created by using two arrays.

- Syntax
- $\circ\;$ The following syntax of the map function is used to return an empty map:
- map ()
 The following syntax of the map function is used to return a map that is created by using two arrays:

```
map(x,y)
```

• Parameters

Parameter	Description
x	The value of this parameter is of the array type.
У	The value of this parameter is of the array type.

Return value type

The map type.

- Examples
- Example 1: The class field specifies classes. The number field specifies the number of students in the classes. The values of the two fields are of
 the array type. Use the map function to create a map based on the values of the two fields. In the returned result, each class is mapped to the
 number of students in the class.
 - Sample fields

class:["class01","class02","class03","class04","class05"]
number:[49,50,45,47,50]

- Query statement
 - * | SELECT map(try_cast(json_parse(class) AS array(varchar)) ,try_cast(json_parse(number) AS array(bigint)))

•	Query	and	analysis	result
---	-------	-----	----------	--------

_col0	\$ Q
{"class01":49,"class03":45,"class02":50,"class05":50,"class04":47}	A
xample 2: Return an empty map.	
Query statement	
* SELECT map()	
Query and analysis result	
_col0	\$ Q
n	

map_agg function

The map_agg function is used to return a map that is created by using x and y. x is a key in the map. y is the value of the key in the map. If y has multiple values, a random value is extracted as the value of the key.

- Syntax
 - map_agg(x, y)
- Parameters

Parameter	Description
X	The value of this parameter is of an arbitrary data type.
у	The value of this parameter is of an arbitrary data type.

Return value type

The map type.

• Example

Extract the values of the **request_method** and **request_time** fields and then use the extracted values to create a map. The value of the **request_method** field is a key in the map. The value of the **request_time** field is the value of the key in the map.

Sample fields

<pre>request_method:POST request_time:80 • Query statement • SELECT map_agg(request_method, request_time) • Query and analysis result col0</pre>			
<pre>request_time:80 • Query statement * SELECT map_agg(request_method, request_time) • Query and analysis result col0</pre>		request_method:POST	
 Query statement SELECT map_agg(request_method, request_time) Query and analysis result col0 Query ind analysis result map_concat function 		request time:80	
 Query statement SELECT map_agg(request_method, request_time) Query and analysis result _col0 Query and analysis result Tecol0 Query and analysis result 			
<pre>* SELECT map_agg(request_method, request_time) • Query and analysis result _col0</pre>	0 1	Ouery statement	
<pre>* SELECT map_agg(request_method, request_time) • Query and analysis result _col0</pre>			
 Query and analysis result _col0		* SELECT map agg(request method, request time)	
• Query and analysis result -col0 ("HEAD":47.0,"DELETE":26.0,"POST":80.0,"GET":51.0,"PUT":49.0) map_concat function			
_col0	0	Ouery and analysis result	
_col0			
{"HEAD":47.0,"DELETE":26.0,"POST":80.0,"GET":51.0,"PUT":49.0} map_concat function		_colu = 0.	
("HEAD":47.0,"DELETE":26.0,"POST":80.0,"GET":51.0,"PUT":49.0} map_concat function			
map_concat function		{"HEAD":47.0,"DELETE":26.0,"POST":80.0,"GET":51.0,"PUT":49.0}	
map_concat function			
map_concat function	-	an concat function	
	ma	ap_concat function	

The map_concat function is used to return the union of multiple maps.

Syntax

map_concat(x, y)

• Parameters

Parameter	Description
x	The value of this parameter is of the map type.
у	The value of this parameter is of the map type.

Return value type

The map type.

• Example

In a log that is transformed by a data transformation job, the values of the **etl_context** and **progress** fields are of the map type. You can use the map_concat function to obtain the union of the field values.

Sample fields

```
etl_context: {
 project:"datalab-148****6461-cn-chengdu"
 logstore:"internal-etl-log"
consumer_group:"etl=83****4d1965"
consumer:"etl=b2d40ed****c8d6-291294"
shard_id:"0" }
progress: {
 accept:3
dropped:0
 delivered:3
 failed:0 }
```

Query statement

```
* |
SELECT
  map concat(
    cast (
     json_parse(etl_context) AS map(varchar, varchar)
    ),
    cast (json_parse(progress) AS map(varchar, varchar))
  )
```

• Query and analysis result

_col0	\$ Q
("consumer_group":"etl-8	974d1965","shard_id":"0","dropped":"0","project"
atalab-148 i6461-cn-chengdu","c	lelivered":"5","failed":"0","logstore":"internal-etl-log","consul
r":"etl-b2d40¢	7a9532c8d6-291294","accept":"5"} Hide

Example

map_filter function

The map_filter function is used to filter elements in a map based on a lambda expression.

Syntax

map_filter(x, lambda_expression)

• Parameters

Parameter	Description
X	The value of this parameter is of the map type.
lambda_expression_expression	The lambda expression. For more information, seeLambda expressions.

Return value type

The map type.

- Example
 - $\label{eq:create} Create a map that does not contain null values from two arrays by using the lambda expression (k, v) -> v is not null .$
 - Query statement
 - * | SELECT map_filter(map(array[10, 20, 30], array['a', NULL, 'c']), (k, v) -> v is not null)

0	Query and analysis result		
	_col0	\$ Q,	
	{"10":"a","30":"c"}		

map_keys function

The map_keys function is used to return an array that consists of all keys in a map.

Syntax

<pre>map_keys(x)</pre>	
Parameters	
Parameter	Description
x	The value of this parameter is of the map type.

Return value type

The array type.

Example

In a log that is transformed by a data transformation job, the value of the **etl_context** field is of the map type. You can use the map_keys function to obtain all keys from the value of the **etl_context** field.

• Sample field

etl_context: {
project:"datalab-148****6461-cn-chengdu"
logstore:"internal-etl-log"
consumer_group:"etl-83****4d1965"
consumer:"etl-b2d40ed****c8d6-291294"
<pre>shard_id:"0" }</pre>

Query statement

* | SELECT map_keys(try_cast(json_parse(etl_context) AS map(varchar, varchar)))

• Query and analysis result

_col0	\$ Q
["consumer", "consumer_group", "logstore", "project", "shard_id"]	•

map_values function

- The map_values function is used to return an array that consists of all values in a map.
- Syntax

map_values(x)

Parameters

Parameter

Description

Cloud Defined Storage

	x	The value of this parameter is of the map type.		
R	eturn value type			
Т	he array type.			
E	xample			
In to o	In a log that is transformed by a data transformation job, the value of the etl_context field is of the map type. You can use the map_values function to obtain the values of all keys from the value of the etl_context field. • Sample field			
	<pre>etl_context: { project:"datalab-148****6461-cn-chengdu" logstore:"internal-etl-log" consumer_group:"etl-83****4d1965" consumer:"etl-b2d40ed****c8d6-291294" shard_id:"0" }</pre>			
Query statement				
	* SELECT map_values(try_cast(json_parse(et)	_context) AS map(varchar, varchar)))		
0	 Query and analysis result 			
	_col0	¢ Q.		
	["etl-85d' f85840-834336	","eti-1143		

multimap_agg function

c","rds_log","datalab-148 66461-cn-chengdu","0"] Hide

The multimap_agg function is used to return a multimap that is created by using x and y. x is a key in the multimap. y is the value of the key in the multimap. The value is of the array type. If y has multiple values, all the values are extracted as the values of the key.

Syntax

multimap_agg(x, y)

• Parameters

Parameter	Description
x	The value of this parameter is of an arbitrary data type.
у	The value of this parameter is of an arbitrary data type.

- Return value type
- The map type.
- Example

Extract all values of the **request_method** and **request_time** fields and then use the extracted values to create a multimap. The value of the **request_method** field is a key in the multimap. The value of the **request_time** field is the value of the key in the multimap. The value of the key is of the array type.

• Sample field

request_method:POST request_time:80

Query statement

* | SELECT multimap_agg(request_method,request_time)

• Query and analysis result

```
_col0
```

4.4.10.4. Approximate functions

This topic describes the syntax of approximate functions. This topic also provides examples on how to use the functions.

Function	Description	Example
<pre>approx_distinct(x)</pre>	Estimates the number of unique values in the x field.	None
<pre>approx_percentile(x,percentage)</pre>	Sorts the values of the x field in ascending order and returns the value that is approximately at the percentage position.	<code>approx_percentile(x,0.5)</code> : returns the value that is approximately at the 50% position in the x field.

<pre>approx_percentile(x, percentages)</pre>	This function is similar to approx_percentile(x,percentage) . You can specify multiple percentages to return the values at the specified percentage positions.	<pre>approx_percentile(x,array[0.1,0.2])</pre>
numeric_histogram(buckets, Value)	Distributes all values of the Value field to multiple buckets. The buckets parameter specifies the number of buckets. The key of each bucket and the number of values in a bucket are returned. This function is equivalent to select count group by . Note The returned result is of the JSON type.	<pre>method:POST select numeric_histogram(10,latency) : distributes the values of the latency field for POST requests to 10 buckets and calculates the number of latency field values in each bucket.</pre>
numeric_histogram_u(buckets, Value)	Distributes all values of the Value field to multiple buckets. The buckets parameter specifies the number of buckets. The key of each bucket and the number of values in a bucket are returned. This function is equivalent to select count group by . Note The returned result is a table that includes multiple rows and columns.	<pre>method:POST select numeric_histogram(10,latency) : distributes the values of the latency field for POST requests to 10 buckets and calculates the number of latency field values in each bucket.</pre>

? Note Buckets are evenly divided by aggregation degree. The returned result for each bucket includes the average value of the bucket and the number of values in the bucket.

4.4.10.5. Mathematical statistics functions

This topic describes the syntax of mathematical statistics functions. This topic also provides examples on how to use the functions.

Syntax

Function	Description
corr(<i>key1, key2</i>)	Calculates the correlation coefficient between two specific columns. The return value is in the range of [0,1].
covar_pop(<i>key1, key2</i>)	Calculates the population covariance of two specific columns.
covar_samp(<i>key1, key2</i>)	Calculates the sample covariance of two specific columns.
regr_intercept(<i>key1, key2</i>)	Returns the linear regression intercept of input values. $key1$ is the dependent value and $key2$ is the independent value.
regr_slope(<i>key1, key2</i>)	Returns the linear regression slope of input values. key1 is the dependent value and key2 is the independent value.
stddev(<i>key</i>)	Calculates the sample standard deviation of the key column. This function is equivalent to the stddev_samp function.
stddev_samp(<i>key</i>)	Calculates the sample standard deviation of the key column.
stddev_pop(<i>key</i>)	Returns the population standard deviation of the key column.
variance(<i>key</i>)	Calculates the sample variance of the key column. This function is equivalent to the var_samp function.
var_samp(<i>key</i>)	Calculates the sample variance of the key column.
var_pop(<i>key</i>)	Calculates the population variance of the key column.

Examples

• Example 1: Calculate the correlation coefficient of two specific columns.

- Query statement
 - * | SELECT corr(request_length, request_time)

Query and analysis result

_col0	\$ Q
0.0008096234574114261	

• Example 2: Query the sample standard deviation and population standard deviation of pre-tax income.

Query statement

* | SELECT stddev(PretaxGrossAmount) as "sample standard deviation", stddev_pop(PretaxGrossAmount) as "population standard deviation", tim e_series(__time__, 'lm', '%H:% I:%s', '0') AS time GROUP BY time



4.4.10.6. Mathematical calculation functions

This topic describes the syntax of mathematical calculation functions. This topic also provides examples on how to use the functions. **Syntax**

? Note

- Mathematical calculation functions support the following operators: $\ensuremath{\ \ +-\star/\ensuremath{\ \ \ }}$.
- In the following functions, x and y can be numbers, log fields, or expressions whose calculation result is a number.

Function	Description
abs(x)	Calculates the absolute value of a number.
cbrt(x)	Calculates the cube root of a number.
sqrt(x)	Calculates the square root of a number.
cosine_similarity(x,y)	Calculates the cosine similarity between x and y.
degrees(x)	Converts radians to degrees.
radians(x)	Converts degrees to radians.
e()	Returns Euler's number.
exp(x)	Returns Euler's number raised to the power of a number.
ln(x)	Calculates the natural logarithm of a number.
log2(x)	Calculates the base-2 logarithm of a number.
log10(x)	Calculates the base-10 logarithm of a number.
log(x,b)	Calculates the base-b logarithm of a number.
pi()	Returns the value of $\boldsymbol{\pi}$ to 14 decimal places.
pow(x,b)	Calculates the value of a number raised to the power of b.
rand()	Returns a random number.
random(0,n)	Returns a random number that is greater than or equal to 0 and less than n.
round(x)	Returns a number rounded to the nearest integer.
round(x, N)	Returns a number rounded to N decimal places.
floor(x)	Returns a number rounded down to the nearest integer. For example, when you execute the * SELECT floor(2.5) statement, 2.0 is returned.
ceiling(x)	Returns a number rounded up to the nearest integer. For example, when you execute the * SELECT ceiling(2.5) statement, 3.0 is returned.
from_base(varchar, bigint)	Converts a string to a base-encoded number.
to_base(x, radix)	Converts a number to a base-encoded string.
truncate(x)	Truncates the fractional part of a number.
acos(x)	Calculates the arc cosine of a number.
asin(x)	Calculates the arc sine of a number.
atan(x)	Calculates the arc tangent of a number.
atan2(y,x)	Calculates the arc tangent of the quotient of a number divided by another number.
cos(x)	Calculates the cosine of a number.
sin(x)	Calculates the sine of a number.

cosh(x)	Calculates the hyperbolic cosine of a number.
tan(x)	Calculates the tangent of a number.
tanh(x)	Calculates the hyperbolic tangent of a number.
infinity()	Returns a positive infinity value.
is_nan(x)	Checks whether a value is a non-numeric value.

Example

Compare the number of page views (PVs) of today with the number of PVs of the previous day, and show the comparison result as a percentage.

• Query statement

* | SELECT diff [1] AS today, round((diff [3] -1.0) * 100, 2) AS growth FROM (SELECT compare(pv, 86400) as diff FROM (SELECT COUNT(*) as pv FROM website_log))

• Query and analysis result

today 🗘 🌣 🔍	growth ¢ ⊂	
1564075.0	-22.11	

4.4.10.7. String functions

This topic describes the syntax of string functions. This topic also provides examples on how to use the functions.

Syntax

? Note

- If you want to use strings in analytic statements, you must enclose the strings in single quotation marks ("). Strings that are not enclosed or enclosed in double quotation marks ("") indicate field names or column names. For example, 'status' indicates the status string, and status or "status" indicates the status log field.
- The **key** parameter in the following table indicates a log field.

Function	Description
chr(<i>number</i>)	Returns the characters that match the ASCII value of a specified parameter.
codepoint(<i>key</i>)	Converts a field of the ASCII type to a field value of the bigint type.
length(<i>key</i>)	Calculates the length of a string. The return value is of the integer type.
lower(<i>key</i>)	Converts the characters in a string to lowercase letters. The return value is of the varchar type in lowercase letters.
upper(<i>key</i>)	Converts the characters in a string to lowercase letters. The return value is of the varchar type in uppercase letters.
lpad(<i>key, length, lpad_string</i>)	 Pads a string to a specified length from the left with a specified substring. The value of the <i>length</i> parameter is an integer that specifies the length of the result string. If the length of the string is less than the value of the<i>length</i> parameter, the string is padded by the specified substring from the left. If the length of the string is greater than the value of the<i>length</i> parameter, the function returns only the first <i>length</i> characters in the string. The return value is of the varchar type.
rpad(<i>key, length,rpad_string</i>)	 Pads a string to a specified length from the right with a specified substring. The value of the <i>length</i> parameter is an integer that specifies the length of the result string. If the length of the string is less than the value of the<i>length</i> parameter, the string is padded by the specified substring from the right. If the length of the string is greater than the value of the<i>length</i> parameter, the function returns only the first <i>length</i> characters in the string. The return value is of the varchar type.
trim(<i>key</i>)	Deletes space characters from the start and the end of a string. The return value is of the varchar type.
ltrim(<i>key</i>)	Deletes space characters from the start of a string. The return value is of the varchar type.
rtrim(<i>key</i>)	Deletes space characters from the end of a string. The return value is of the varchar type.
replace(<i>key,substring,replace</i>)	Replaces matched characters in a string with specified characters. The return value is of the varchar type.

replace(<i>key,substring</i>)	Deletes matched characters from a string. The return value is of the varchar type.
reverse(<i>key</i>)	Reverses the characters in a string.
split(<i>key,delimeter,M</i>)	Splits a string with a specified delimiter and returns a set of N substrings. The return value is an array.
split_part(<i>key,delimeter,part</i>)	Splits a string with a specified delimiter and returns the substring at a specified position. The value of the <i>part</i> parameter is an integer that is greater than 0. The return value is of the varchar type.
split_to_map(<i>key, delimiter01, delimiter02</i>)	Splits a string with the first specified delimiter, and then splits the string with the second specified delimiter. The return value is a map.
position(<i>substring</i> IN <i>key</i>)	Returns the position of a specified substring in a string. The return value is of the integer type. The value starts from 1.
strpos(<i>key, substring</i>)	Returns the position of a specified substring in a string. This function is equivalent to the position¢ <i>ubstring</i> IN <i>key</i>) function. The return value is of the integer type. The value starts from 1.
substr(<i>key, start</i>)	Returns the substring at a specified position in a string. The <i>start</i> parameter specifies the position of the substring to be extracted. The value of the start parameter starts from 1. The return value is of the varchar type.
substr(<i>key, start, length</i>)	Returns the substring at a specified position in a string and specifies the length of the substring. The <i>start</i> parameter specifies the position of the substring to be extracted. The value of the start parameter starts from 1. The <i>length</i> parameter specifies the length of the substring. The return value is of the varchar type.
concat(<i>key01,key02,key03</i>)	Concatenates multiple strings into one string. The return value is of the integer type. The value starts from 1.
levenshtein_distance(key01, key02)	Returns the minimum edit distance between two strings.
hamming_distance (<i>string1,string2</i>)	Returns the Hamming distance between two strings.

Examples

Sample log:

http_user_agent:Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_5_4; en-us) AppleWebKit/528.4+ (KHTML, like Gecko) Version/4.0dpl Safari/526.11.2 request_uri:/request/path-1/file-9?0457349059345

scheme:https

server_protocol:HTTP/2.0

region:cn-shanghai

time: upstream_response_time:"80", request_time:"40"

• Use a question mark (?) to split the value of the request_uri field and return the first substring. The returned substring indicates a file path. Then, calculate the number of requests that correspond to each path.

* | SELECT count(*) AS PV, split_part(request_uri, '?', 1) AS Path GROUP BY Path ORDER BY pv DESC LIMIT 3

PV ‡ Q	Path ‡ Q
49	/request/path-2/file-6
47	/request/path-2/file-0
44	/request/path-3/file-2

• Extract the first four characters (HTTP) from the value of the server_protocol field and calculate the number of requests that use the HTTP protocol.

 \star | SELECT substr(server_protocol,1,4) AS protocol, count(*) AS count GROUP BY server_protocol

protocol 🗘 🗘	count 🗘	2
НТТР	9078	

• Use commas (,) and colons (:) to split the value of the time field and return a value of the map type.

* | SELECT split_to_map(time,',',':')

_col0	\$Q
{"request_time":"\"40\"","upstream_response_time":"\"80\""}	
{"request_time":"\"40\"","upstream_response_time":"\"80\""}	

• Check whether the value of the http_user_agent field starts with the letter M.

*	SELECT	substr(http	_user_agent,	1,	1)=chr(77)
---	--------	-------------	--------------	----	------------

_col0	\$
rue	
rue	

• Return the position of the letter H in the value of the **server_protocol** field.

* | SELECT strpos(server_protocol,'H')

_col0	÷	٩
1		
1		

• Use a forward slash (/) to split the value of the server_protocol field into two substrings and return an array of the substrings.

_col0)	

_col0	\$ Q.
["HTTP","2.0"]	^
eplace cn in the region field with China .	
<pre>* select replace(region,'cn','China')</pre>	
_col0	\$ Q
China-shanghai	A

China-shanghai

٠

4.4.10.8. Date and time functions

Log Service provides the following types of date and time functions that you can use to analyze log data: date function, time function, truncation function, interval function, and time series padding function. You can use the functions to convert the date and time formats of log data. You can also use the functions to group and aggregate log data.

? Note

- The timestamp of a log in Log Service is accurate to seconds. Therefore, you can specify the precision of the time format only to seconds.
- You need to specify the time format only for the time in a time string. Other parameters such as the time zone are not required.
- Each log in Log Service contains the reserved _time_ field. The value of the field is a UNIX timestamp. For example, 1592374067 indicates 2020-06-17 14:07:47.

Date functions

Function	Description	Example	
current_date	Returns the current date. Return value format: YYYY-MM-DD Return value type: date. 	* select current_date	
current_time	Returns the current time. Return value format: HH:MM:SS.Ms Time zone Example: 01:14:51.967 Asia/Shanghai. Return value type: time.	* select current_time	
current_timestamp	 Returns the current date and time. Return value format: YYYY-MM-DD HH:MM:SS.Ms Time zone . Example: 2021-01-12 17:16:09.035 Asia/Shanghai. Return value type: timestamp. 	* select current_timestamp	
current_timezone()	Returns the current time zone. Return value type: varchar. Example: Asia/Shanghai.	<pre>* select current_timezone()</pre>	
localtime	Return sthe local time. • Return value format: HH:MM:SS.Ms • Return value type: time.	* select localtime	
localtimestamp	Return value format: YYYY-MM-DD HH:MM:SS.Ms • Return value type: timestamp.	* select localtimestamp	

Cloud Defined Storage

User Guide-Log Service

now()	Returns the current date and time. This function is equivalent to the current_timestamp function. Return value format: YYYY-MM-DD HH:MM:SS.Ms Time zone Return value type: timestamp.	<pre>* select now()</pre>
from_iso8601_timestamp(<i>\SO8601</i>)	Converts an ISO 8601-formatted datetime expression to a timestamp expression that contains a time zone. • Return value format: YYYY-MM-DD HH:MM:SS.Ms Time zone . • Return value type: timestamp.	<pre>* select from_iso8601_timestamp('2020-05- 03T17:30:08')</pre>
from_iso8601_date(<i>ISO8601</i>)	Converts an ISO 8601-formatted date expression to a date expression. • Return value format: YYYY-MM-DD • Return value type: date.	* select from_iso8601_date('2020- 05-03')
from_unixtime(UNIX timestamp)	Converts a UNIX timestamp to a timestamp expression. Return value format: YYYY-MM-DD HH:MM:SS.Ms Return value type: timestamp. 	* select from_unixtime(1494985275)
from_unixtime(UNIX timestamp,time zone)	Converts a UNIX timestamp to a timestamp expression that contains a time zone. • Return value format: YYYY-MM-DD HH:MM:SS.Ms Time zone . • Return value type: timestamp.	* select from_unixtime (1494985275,'Asia/Shanghai')
to_unixtime(<i>timestamp</i>)	Converts a timestamp expression to a UNIX timestamp. Return value type: long. Example: 1494985500.848.	* select to_unixtime('2017-05-17 09:45:00.848 Asia/Shanghai')

Time functions

Function	Description	Example
date_format(<i>timestamp,format</i>)	Converts a timestamp expression to a datetime format string.	* select date_format (date_parse('2017- 05-17 09:45:00','%Y-%m-%d %H:%i:%S'), '%Y- %m-%d')
date_parse(<i>string,format</i>)	Represents a datetime format string, and then converts the datetime format string to a timestamp expression. The following table describes the formats.	<pre>* select date_format (date_parse(time,'%Y-%m-%d %H:%i:%S'), '%Y-%m-%d')</pre>

Table 1. Formats

format	Description
%a	The abbreviated name of the day of the week. Examples: Sun and Sat.
%b	The abbreviated name of the month. Examples: Jan and Dec.
%с	The month. The value is of the numeric type. Valid values: 1 to 12.
%D	The day of the month. Examples: 0th, 1st, 2nd, and 3rd.
%d	The day of the month. The value is in the decimal format. Valid values: 01 to 31.
%e	The day of the month. The value is in the decimal format. Valid values: 1 to 31.
%H	The hour. The 24-hour clock is used.
%h	The hour. The 12-hour clock is used.
%I	The hour. The 12-hour clock is used.
%i	The minute. The value is of the numeric type. Valid values: 00 to 59.
%j	The day of the year. Valid values: 001 to 366.
%k	The hour. Valid values: 0 to 23.
%I	The hour. Valid values: 1 to 12.
%M	The full month name. Examples: January and December.
%m	The month. The value is of the numeric type. Valid values: 01 to 12.
%p	The abbreviation that indicates the morning or afternoon of the day. Valid values: AM and PM.
%r	The time. The 12-hour clock is used. The time is in the $hh:mm:ss AM/PM$ format.
%S	The second. Valid values: 00 to 59.
%s	The second. Valid values: 00 to 59.

%Т	The time. The 24-hour clock is used. The time is in the hh:mm:ss format.
%V	The week number of the year. Sunday is the first day of each week. Valid values: 01 to 53.
%v	The week number of the year. Monday is the first day of each week. Valid values: 01 to 53.
%W	The full name of the day of the week. Examples: Sunday and Saturday.
%w	The day of the week. The value 0 indicates Sunday.
%Y	The four-digit year. Example: 2020.
%у	The two-digit year. Example: 20.
%%	The escape character of the percent sign (%).

Truncation function

The date_trunc() function truncates a datetime expression based on the specified part of a time. You can use a truncation function to truncate a time by second, minute, hour, day, month, or year. This function is suitable for time-based statistics.

Syntax

date_trunc('unit',x)

• Parameters

The value of the x parameter can be a datetime expression, for example, 2021-01-12 03:04:05.000 or 1610350836. The value of the x parameter can be a time field, for example, __time__. The valid values of the unit parameter are second, minute, hour, day, week, month, quarter, and year. The following table describes examples of this parameter.

Example	Result	Description
* select date_trunc('second', 2021-01-12 03:04:05.000)	2021-01-12 03:04:05.000	None.
* select date_trunc('minute', 2021-01-12 03:04:05.000)	2021-01-12 03:04:00.000	None.
* select date_trunc('hour', 2021-01-12 03:04:05.000)	2021-01-12 03:00:00.000	None.
* select date_trunc('day', 2021-01-12 03:04:05.000)	2021-01-12 00:00:00.000	Returns 00:00:00.000 of the specified date.
* select date_trunc('week', 2021-01-12 03:04:05.000)	2021-01-11 00:00:00.000	Returns 00:00:00.000 of the Monday of the specified week.
* select date_trunc('month', 2021-01-12 03:04:05.000)	2021-01-01 00:00:00.000	Returns 00:00:00.000 of the first day of the specified month.
* select date_trunc('quarter', 2021-01-11 03:04:05.000)	2021-01-01 00:00:00.000	Returns 00:00:00.000 of the first day of the specified quarter.
* select date_trunc('year', 2021-01-11 03:04:05.000)	2021-01-01 00:00:00.000	Returns 00:00:00.000 of the first day of the specified year.

• Query and analysis examples

To truncate the average request durations by minute, and group and sort the average request durations by time, execute the following query statement:

```
* | select date_trunc('minute', __time__) as time,
    truncate (avg(request_time) ) as avg_time ,
    current_date as date
    group by time
    order by time desc
    limit 100
```

You can use the date trunc('unit', x) function to truncate a time only by second, minute, hour, day, week, month, quarter, or year. To truncate a time based on specified intervals such as 5 minutes, you must use a GROUP BY clause based on the modulus method.

* | select count(1) as pv, __time__ - __time__ %300 as time group by time limit 100

In the preceding statement, \$300 indicates that modulo and truncation are performed every 5 minutes.

Interval functions

You can use interval functions to perform interval-related calculations. For example, you can add or subtract an interval based on a date, or calculate the interval between two dates.

Function	Description	Example
date_add(<i>unit, N,timstamp</i>)	Adds N units to a timestamp . To subtract an interval, set the value of <i>N</i> to a negative value.	<pre>*! select date_add('day', -7, '2018-08-09 00:00:00') Indicates seven days before August 9, 2018 (2018- 08-02 00:00:00.000).</pre>
date_diff(<i>unit, timestamp1, timestamp2</i>)	Returns the time difference between two time expressions. For example, you can calculate the difference between timestamp1 and timestamp2 by unit.	<pre>* select date_diff('day', '2018-08-02 00:00:00', '2018-08-09 00:00:00')</pre>

The following table describes the valid values of the unit parameter.

unit

Description

Cloud Defined Storage

millisecond	milliseconds
second	seconds
minute	minutes
hour	hours
day	days
week	weeks
month	months
quarter	quarters
year	years

Time series padding function

You can use the time_series() function to add the missing data when you query in the time window.

() Important You must use the time_series() function together with GROUP BY and ORDER BY clauses. You cannot use the DESC keyword in an ORDER BY clause to sort data returned in descending order.

• Syntax

time_series(time_column, window, format, padding_data)

• Parameters

Parameter	Description
time_column	The sequence of time (KEY), for example, time The value of this parameter can be a long datetime or timestamp expression.
window	The size of the window. Valid units: s (seconds), m (minutes), h (hours), and d (days). Examples: 2h, 5m, and 3d.
format	The format in which you want the function to return the value.
padding_data	 The content that you want to add. Valid values: 0: adds 0. null: adds null. last: adds the value of the previous point in time. next: adds the value of the next point in time. avg: adds the average of the value of the previous point in time and the value of the next point in time.

Example

To add missing data by 2 hours, execute the following query statement:

* | select time_series(__time_, '2h', '%Y-%m-%d %H:%i:%s', '0') as time, count(*) as num from log group by time order by time

time 💠 🔍	num \$\$ 0.
2021-07-20 00:00:00	11602
2021-07-20 02:00:00	63089
2021-07-20 04:00:00	36583
2021-07-20 06:00:00	11135
2021-07-20 08:00:00	62746
2021-07-20 10:00:00	18314

4.4.10.9. URL functions

This topic describes the syntax of URL functions. This topic also provides examples on how to use the functions.

URL functions extract fields from standard URLs. The following example shows the format of a URL:

[protocol:][//host[:port]][path][?query][#fragment]

The following table describes common URL functions.

Eurotion	Description	Example	
Function	Description	Query statement	Query result
<pre>url_extract_fragme nt(url)</pre>	Extracts the fragment from a URL. The return value is of the varchar type.	<pre>* select url_extract_fragment('https://sls.console.aliyun.com/#/project/das board-demo/categoryList')</pre>	/project/dashboard h- demo/categoryList

url_extract_host(u rl)	Extracts the host from a URL. The return value is of the varchar type.	<pre>* select url_extract_host('http://www.aliyun.com/product/sls')</pre>	www.aliyun.com
<pre>url_extract_parame ter(url, name)</pre>	Extracts the value of a specified parameter in the query string from a URL. The return value is of the varchar type.	<pre>* select url_extract_parameter('http://www.aliyun.com/product/sls? userid=testuser','userid')</pre>	testuser
url_extract_path(u rl)	Extracts the path from a URL. The return value is of the varchar type.	<pre>* select url_extract_path('http://www.aliyun.com/product/sls? userid=testuser')</pre>	/product/sls
url_extract_port(url)	Extracts the port number from a URL. The return value is of the bigint type.	<pre>* select url_extract_port('http://www.aliyun.com:80/product/sls? userid=testuser')</pre>	80
<pre>url_extract_protoc ol(url)</pre>	Extracts the protocol from a URL. The return value is of the varchar type.	<pre>* select url_extract_protocol('http://www.aliyun.com:80/product/sls? userid=testuser')</pre>	http
<pre>url_extract_query(url)</pre>	Extracts the query string from a URL. The return value is of the varchar type.	<pre>* select url_extract_query('http://www.aliyun.com:80/product/sls? userid=testuser')</pre>	userid=testuser
url_encode(value)	Encodes a URL.	<pre>* select url_encode('http://www.aliyun.com:80/product/sls? userid=testuser')</pre>	http%3a%2f%2fwww.a liyun.com%3a80%2fpro duct%2fsls%3fuserid% 3dtestuser
url_decode(value)	Decodes a URL.	* select url_decode('http%3a%2f%2fwww.aliyun.com%3a80%2fproduct%2fsls%3fuse id%3dtestuser')	http://www.aliyun. ercom:80/product/sls? userid=testuser

4.4.10.10. Regular expression functions

This topic describes the available regular expression functions. You can use these functions when you query and analyze data in Log Service.

A regular expression function parses a string and returns the required substrings.

The following table lists common regular expression functions.

Function	Description	Example
<pre>regexp_extract_all(string, pattern)</pre>	Returns an array where each element is a substring that matches the regular expression. These substrings derive from the specified string.	The returned result of * SELECT regexp_extract_all('5a 67b 890m', '\d+') is ['5','67','890'] . The returned result of * SELECT regexp_extract_all('5a 67a 890m', '(\d+)a') is ['5a','67a'] .
<pre>regexp_extract_all(string, pattern, group)</pre>	Returns an array where each element is a part of a substring that matches the regular expression. This part is the content in the group parameter value of the () of a substring that derives from the specified string.	The returned result of * SELECT regexp_extract_all('5a 67a 890m', '(\d+)a',1) is ['5','67'] .
<pre>regexp_extract(string, pattern)</pre>	Returns the first substring of the specified string that matches the regular expression.	The returned result of * SELECT regexp_extract('5a 67b 890m', '\d+') is '5'.
<pre>regexp_extract(string, pattern,group)</pre>	Returns a part of the first substring that matches the regular expression. This part is the content in the group parameter value of the $()$ of the substring that derives from the specified string.	The returned result of * SELECT regexp_extract('5a 67b 890m', '(\d+)([a-z]+)',2) is 'a'.
<pre>regexp_like(string, pattern)</pre>	Returns a Boolean value. If the string and its substrings cannot match the regular expression, the value False is returned.	The returned result of * SELECT regexp_like('5a 67b 890m', '\d+m') is True.
<pre>regexp_replace(string, pattern, replacement)</pre>	Replaces the substrings of the specified string that match the regular expression with the value of the replacement parameter.	The returned result of * SELECT regexp_replace('5a 67b 890m', '\d+','a') is 'aa ab am' .
<pre>regexp_replace(string, pattern)</pre>	Removes the substrings of the specified string that match the regular expression. This function is equivalent to regexp_replace(string,patterm,'') .	The returned result of * SELECT regexp_replace('5a 67b 890m', '\d+') is 'a b m' .
<pre>regexp_split(string, pattern)</pre>	Returns an array where each element is a substring of the specified string that is split based on the regular expression.	The returned result of * SELECT regexp_split('5a 67b 890m', '\d+') is ['a','b','m'] .

4.4.10.11. JSON functions

This topic describes the syntax of JSON functions. This topic also provides examples on how to use the functions.

? Note

- If a string fails to be parsed into JSON data, null is returned.
- In analytic statements of Log Service, a JSON array that is enclosed in single quotation marks (") indicates a string.
- If the value of a log field is of the JSON type and needs to be expanded to multiple rows, we recommend that you use UNNEST clauses. For more information, see UNNEST clause.

json_parse() function

The json_parse() function is used to convert a string to JSON data. The returned result is of the JSON type.

Syntax

json_parse(string)

• Example

Convert the [1, 2, 3] string to the [1,2,3] JSON array.

* | SELECT json_parse('[1, 2, 3]')

The returned result is [1,2,3].

json_format()

- The json_format() function is used to convert JSON data to a string. The returned result is a string.
- Syntax

```
json_format(json)
```

• Example

Convert the [1,2,3] JSON array to the [1, 2, 3] string.

* | SELECT json_format(json_parse('[1, 2, 3]'))

The returned result is [1,2,3].

json_array_contains()

The json_array_contains() function is used to check whether a JSON array or a JSON string contains a specified value. The returned result is true or false.

Svntax

```
json_array_contains(json , value)
```

• Examples

```
• Check whether the [1, 2, 3] JSON array contains 2.
```

* | SELECT json_array_contains(json_parse('[1, 2, 3]'), 2)

The returned result is true.

```
\circ~ Check whether the [1, 2, 3] JSON string contains 2.
```

* | SELECT json_array_contains('[1, 2, 3]', 2)

The returned result is true.

json_array_get()

The json_array_get() function is used to extract the element that corresponds to the subscript of a JSON array.

Syntax

json_array_get(json_array, index)

• Example

```
Extract the element that corresponds to the subscript 0 of the ["status", "request_time", "request_method"] JSON array.
```

* | SELECT json_array_get('["status", "request_time", "request_method"]', 0)

The returned result is status.

json_array_length()

The json_array_length() function is used to calculate the number of elements in a JSON array.

Syntax

json_array_length(json array)

• Example

Calculate the number of the elements in the ["status", "request_time", "request_method"] JSON array.

```
* | SELECT json_array_length('["status", "request_time", "request_method"]')
```

The returned result is 3.

json_extract()

The json_extract() function is used to extract the value of a specified field from a JSON object. The returned result is of the JSON type.

③ Note If the JSON data is invalid when you use the json_extract() function, an error message appears. We recommend that you use the json_extract_scalar() function.

• Syntax

> Document Version: 20240703

```
json_extract(json, json_path)
```

The format of json_path is \$.store.book[0].title .

- Examples
 - Extract the value of the status field from the content field. The content field is a JSON object.
 - * | SELECT json extract(content, '\$.status')

The returned result is the value of the **status** field, for example, "200".

 Expand the value of the request_time field and use row to represent the expanded rows. The value of the request_time field is a JSON array. Then, extract and calculate the sum of the values of the status field from the rows.

* | select sum(cast (json_extract_scalar(row, '\$.status') as bigint)) from log, unnest(cast(json_parse(request_time) as array(json)) a s t(row)

The returned result is the sum result.

json_extract_scalar()

The json_extract_scalar() function is used to extract the value of a specified field from a JSON object. The returned result is a string.

```
    Syntax
```

json_extract_scalar(json, json_path)

The format of json_path is \$.store.book[0].title .

- Examples
 - Extract the value of the status field from the content field. The content field is a JSON object.

* | SELECT json_extract_scalar(content, '\$.status')

The returned result is the value of the status field, for example, "200".

• Extract the value of the status field from the content field. The content field is a JSON object. Then, convert the value to the bigint type and calculate the sum.

* | select sum(cast (json_extract_scalar(content, '\$.status') as bigint))

The returned result is the sum result.

json_size()

The json_size() function is used to calculate the number of elements in a JSON object or JSON array.

```
    Syntax
```

json_size(json,json_path)

Example

Calculate the number of elements in the status field.

* | SELECT json_size('{"status":[1, 2, 3]}','\$.status')

The returned result is 3.

4.4.10.12. Type conversion functions

Type conversion functions convert the data type of a specified value or column in a query.

You can use the index attribute feature of Log Service to set the data type of a field to LONG, DOUBLE, TEXT, or JSON. You can also query fields of various data types, including BIGINT, DOUBLE, VARCHAR, and TIMESTAMP. To query fields of a specific data type, you can use type conversion functions to convert the data type configured in an index into the data type that you use in a query.

Syntax

(2) Note We recommend that you use the try_cast() function if a log contains dirty data. Otherwise, a query may fail due to the dirty data.

• Convert the data type of a column of values or a specific value into the specified type in a query. If the data type of a value fails to be converted, the query is terminated.

cast([key|value] AS type)

• Convert the data type of a column of values or a specific value into the specified type in a query. If the data type of a value fails to be converted, NULL is returned for the value, and the query continues.

try_cast([key|value] AS type)

Parameter	Description
key	The key of a field whose value data type is to be converted.
value	The field value whose data type is to be converted into the specified type.

Example

- To convert the numeric value 123 to a value of the VARCHAR type, use the following statement: cast (123 AS varchar)
- To convert the data type of the uid field values to the VARCHAR type, use the following statement: cast(uid AS varchar)

4.4.10.13. IP functions

IP functions can be used to identify whether an IP address is an internal or external IP address. IP functions can also be used to identify the country, state, and city to which an IP address belongs. This topic describes the syntax of IP functions and provides examples on how to use the functions.

③ Note The KEY parameter in the following functions indicates a log field, for example, client_ip. The value of this parameter is an IP address.

Function	Description	Example
ip_to_domain(KEY)	 Checks whether an IP address is an internal IP address or an external IP address. The returned result is intranet or internet. intranet: an internal IP address. internet: an external IP address. 	<pre>* SELECT ip_to_domain(client_ip)</pre>
ip_to_country(KEY)	Identifies the country or the region to which an IP address belongs. The returned result is the Chinese name of a country or a region.	<pre>* SELECT ip_to_country(client_ip)</pre>
ip_to_country(KEY,'en')	Identifies the country or the region to which an IP address belongs. The returned result is the code of a country or a region.	<pre>* SELECT ip_to_country(client_ip,'en')</pre>
ip_to_country_code(KEY)	Identifies the country or the region to which an IP address belongs. The returned result is the code of a country or a region.	<pre>* SELECT ip_to_country_code(client_ip)</pre>
ip_to_province(KEY)	Identifies the state to which an IP address belongs. The returned result is the Chinese name of a state.	* SELECT ip_to_province(client_ip)
ip_to_province(KEY,'en')	Identifies the state to which an IP address belongs. The returned result is the administrative region code of a state.	<pre>* SELECT ip_to_province(client_ip,'en')</pre>
ip_to_city(KEY)	Identifies the city to which an IP address belongs. The returned result is the Chinese name of a city.	<pre>* SELECT ip_to_city(client_ip)</pre>
ip_to_city(KEY,'en')	Identifies the city to which an IP address belongs. The returned result is the administrative region code of a city.	<pre>* SELECT ip_to_city(client_ip, 'en')</pre>
ip_to_geo(KEY)	Identifies the longitude and latitude of the location to which an IP address belongs. The returned result is in the latitude, longitude format. For information about geohash functions, see Geography functions.	* SELECT ip_to_geo(client_ip)
ip_to_city_geo(KEY)	Identifies the longitude and latitude of the city to which an IP address belongs. This function returns the longitude and latitude of a city. Each city has only one set of coordinates. The returned result is in the latitude, longitude format.	<pre>* SELECT ip_to_city_geo(client_ip)</pre>
ip_to_provider(KEY)	Identifies the Internet service provider (ISP) of an IP address. The returned result is the name of an ISP.	* SELECT ip_to_provider(client_ip)

4.4.10.14. GROUP BY clause

GROUP BY clauses are used together with aggregate functions to group analysis results based on one or more columns.

Syntax

 \star | SELECT column name, aggregate function GROUP BY [column name | alias | serial number]

(2) Note If you use a GROUP BY clause in an SQL statement, you can perform aggregate calculations on only a column that is specified in the GROUP BY clause or a random column that is not a non-GROUP BY column. For example, * | SELECT status, request_time, COUNT(*) AS PV GROUP BY status is an invalid query statement because request_time is not a GROUP BY column.

A GROUP BY clause can be used to group data by column name, alias, and serial number. The following table describes the related parameters.

Parameter	Description
Column name	The name of a log field or the return column of an aggregate function. You can group data by log field name (key) or the result of an aggregate function. (key) or the result of an aggregate function. A GROUP BY clause supports single column or multiple columns.

Alias	Data is grouped by the alias of a log field name or the return column alias of an aggregate function. You can specify the alias of a log field in the Field Search section of the Search & Analysis panel. For more information, see Column aliases.	
	The serial number of a column in a SELECT statement. The number starts from 1. For example, the serial number of the status column is 1. In this case, the following two statements are equivalent:	
Serial number	* SELECT status, count(*) AS PV GROUP BY status	
	* SELECT status, count(*) AS PV GROUP BY 1	
Aggregate function	A GROUP BY clause can be used together with aggregate functions such as MIN, MAX, AVG, SUM, and COUNT. For more information, see General aggregate functions.	

Examples

• To calculate the number of access requests of different HTTP status codes, you can execute the following query statement:

* | SELECT status, count(*) AS PV GROUP BY status

• To calculate the number of page views (PVs) by 1 hour, you can execute the following query statement:

 \star | SELECT count(*) AS PV , date_trunc('hour', __time__) AS time GROUP BY time ORDER BY time limit 1000

The _time_ field is a reserved field in Log Service. This field indicates the time column. time is the alias of date_trunc('hour', _time_). For more information about the date_trunc() function, see Truncation function.

⑦ Note

- The clause limit 1000 indicates that a maximum of 1,000 rows of data can be returned. If you do not use a LIMIT clause, you can obtain a maximum of 100 rows of data by default.
- If you turn on Enable Analytics for a log field in the Search & Analysis panel, the analysis feature is automatically enabled for the <u>time</u> field.
- To calculate the number of PVs by 5 minutes, you can execute the following query statement.

The date_trunc() function can only collect statistics at a specified interval. If you want to perform statistical analysis by custom time, you can group data based on the modulus method. In this example, **%300** in the following statement indicates that the modulo and truncation operations are performed every 5 minutes.

* | SELECT count(*) AS PV, __time__ - __time__%300 AS time GROUP BY time limit 1000

• To extract a column that is not grouped by using a GROUP BY clause, you can execute the following query statement. If you use a GROUP BY clause in an SQL statement, you can perform aggregate calculations on only a column that is specified in the GROUP BY clause or a random column that is not a non-GROUP BY column. For example, * | SELECT status, request_time, COUNT(*) AS PV GROUP BY status is an invalid query statement because request_time is not a GROUP BY column.

* | SELECT status, arbitrary(request_time), count(*) AS PV GROUP BY status

4.4.10.15. Window functions

This topic describes the syntax of window functions.

Window functions are used to perform calculations across rows of a log. Common SQL aggregate functions calculate the results of only one row or aggregate all rows into one row for calculation. Window functions support cross-row calculation and fill the calculation results in each row.

```
Syntax of window functions:

SELECT key1, key2, value,

rank() OVER (PARTITION BY key2

ORDER BY value DESC) AS rnk

FROM orders

ORDER BY key1,rnk
```

rank() OVER (PARTITION BY KEY2 ORDER BY value DESC) indicates that PARTITION BY is first used to partition by KEY2 and sort by the value if KEY2 is the same, and then the rank() function is used to aggregate data.

Special aggregate functions used in windows

Function	Description
rank()	Returns the rank of a value within a group of values. The rank is one plus the number of preceding rows that are not peers of the current row.
row_number()	Returns a unique, sequential number for each row.
first_value(x)	Returns the first value in the window. In most cases, the function is used to sort all values in a window and then return the maximum value.
last_value(x)	Returns the last value in the window. In most cases, the function is used to sort all values in a window and then return the minimum value.
nth_value(x, offset)	Returns the value at the specified offset from the beginning of the window.
lead(x,offset,defaut_value)	Returns the value in offset rows that follow the current row in the window. If the target row does not exist, the default_value is returned.
lag(x,offset,defaut_value)	Returns the value in offset rows that precede the current row in the window. If the target row does not exist, the default_value is returned.

Examples

• To rank the salaries of employees in their departments, execute the following query statement:

* | select department, persionId, sallary , rank() over(PARTITION BY department order by sallary desc) as sallary_rank order by department, sallary_rank

Query and analysis result

department	persionId	sallary	sallary_rank
dev	john	9000	1
dev	Smith	8000	2
dev	Snow	7000	3
dev	Achilles	6000	4
Marketing	Blan Stark	9000	1
Marketing	Rob Stark	8000	2
Marketing	Sansa Stark	7000	3

• To calculate the percentages of salaries of employees in their departments, execute the following query statement:

* | select department, persionId, sallary *1.0 / sum(sallary) over(PARTITION BY department) as sallary_percentage
Query and analysis result

department	persionId	sallary	sallary_percentage
dev	john	9000	0.3
dev	Smith	8000	0.26
dev	Snow	7000	0.23
dev	Achilles	6000	0.2
Marketing	Blan Stark	9000	0.375
Marketing	Rob Stark	8000	0.333
Marketing	Sansa Stark	7000	0.29

• To calculate the daily UV increase over the previous day, execute the following query statement:

* | select day ,uv, uv *1.0 /(lag(uv,1,0) over()) as diff_percentage from

select approx_distinct(ip) as uv, date_trunc('day', __time__) as day from log group by day order by day asc

Query and analysis result

day	uv	diff_percentage
2017-12-01 00:00:00	100	null
2017-12-02 00:00:00	125	1.25
2017-12-03 00:00:00	150	1.2
2017-12-04 00:00:00	175	1.16
2017-12-05 00:00:00	200	1.14
2017-12-06 00:00:00	225	1.125
2017-12-07 00:00:00	250	1.11

4.4.10.16. HAVING clause

This topic describes the syntax of HAVING clauses.

The query and analysis feature of Log Service supports the standard SQL HAVING clause. A HAVING clause is used together with a GROUP BY clause to filter GROUP BY results.

The following example shows the syntax of a HAVING clause:

method :PostLogstoreLogs |select avg(latency),projectName group by projectName having avg(latency) > 100

Difference between HAVING and WHERE clauses

A HAVING clause is used to filter the aggregation and calculation results after you use a GROUP BY clause. A WHERE clause is used to filter the raw data during aggregation.

Example

To calculate the average rainfall of each province in which the temperature is higher than 10°C, and return only the provinces in which the average rainfall is greater than 100 ml, execute the following query statement:

 \star | select avg(rain) ,province where temperature > 10 group by province having avg(rain) > 100

4.4.10.17. ORDER BY clause

This topic describes the syntax of ORDER BY clauses.

- An ORDER BY clause is used to sort query results based on only one column.
- Syntax

```
order by column name [desc|asc]
```

Example

```
method :PostLogstoreLogs |select avg(latency) as avg_latency,projectName group by projectName HAVING avg(latency) > 5700000
```

order by avg_latency desc

4.4.10.18. LIMIT syntax

The LIMIT clause is used to limit the number of returned rows.

Syntax formats

Log Service supports the following LIMIT syntax formats:

• Reads only the first N rows:

- limit N
- Reads N rows starting from the S-th row:

limit S , N

? Note

- If you use the LIMIT clause to paginate results, the final results rather than the intermediate results of the SQL query are obtained.
 You cannot apply the LIMIT clause to subqueries. For example, the following statement is not supported:
 - * | select count(1) from (select distinct(url) from limit 0,1000)
- If you use the LIMIT clause for pagination, the offset value cannot exceed 1,000,000. For example, in the limit s , N clause, the sum of S and N cannot exceed 1,000,000, and the value of N cannot exceed 10,000.

Example

- To obtain the first 100 rows of results, run the following statement.
 - * | select distinct(url) from log limit 100
- To obtain a total of 1,000 results from row 0 to row 999, run the following statement.

* | select distinct(url) from log limit 0,1000

- To obtain a total of 1,000 results from row 1,000 to row 1,999, run the following statement:
 - * | select distinct(url) from log limit 1000,1000

4.4.10.19. Conditional expressions

This topic describes the syntax of conditional expressions and provides examples on how to use conditional expressions.

CASE WHEN statement

CASE WHEN statements are used to classify data.

Syntax

```
CASE WHEN condition1 THEN result1
[WHEN condition2 THEN result2]
[ELSE result3]
```

• Examples

 Extract browser information from the value of the http_user_agent field, classify the information into Chrome, Safari, and unknown types, and then calculate the number of page views (PVs) for the three types.

Query statement

* SELECT CASE				
WHEN http_user_agent	like	'%Chrome%'	then	'Chrome'
WHEN http_user_agent	like	'%Safari%'	then	'Safari'
ELSE 'unknown'				
END AS http_user_agent,				
count(*) AS pv				
GROUP BY http_user_agent				

Query and analysis result

http_user_agent 🗘 ್ಲಿ	р и \$0,
Chrome	5563
Safari	1842
unknown	1666

• Query the distribution of requests that are sent at different points in time.

 Query statement 		
* SELECT		
CASE		
WHEN request_time < 10 then 't10'		
WHEN request_time < 100 then 't100'		
WHEN request_time < 1000 then 't1000'		
WHEN request_time < 10000 then 't10000'		
ELSE 'large' END		
AS request_time,		
count(*) AS pv		
GROUP BY request_time		

Query and analysis result

request_time \$ 0	₽∨ \$ <
t100	1563542
large	533

if() function

The if() function is used to classify data. This function is similar to CASE WHEN statements.

• Syntax

• If the *condition* is true, the *true_value* column is returned. Otherwise, null is returned.

if(condition, true_value)

• If the condition is true, the true_value column is returned. Otherwise, the false_value column is returned.

if(condition, true_value, false_value)

• Example

Calculate the ratio of requests whose status code is 200 to all requests.

- Query statement
 - * | SELECT sum(if(status =200,1,0))*1.0 / count(*) AS status_200_percentage
- Query and analysis result

status_200_percentage	\$ Q.
0.8846858366766299	

coalesce() function

The coalesce() function is used to return the first non-null value in multiple columns.

Syntax

coalesce(expression1, expression2, expression3, expression4)

Example

Calculate the ratio of the expenses of yesterday to the expenses of the same day last month.

Query statement

* | SELECT compare("expenses of yesterday", 604800) AS diff FROM (SELECT coalesce(sum(PretaxAmount), 0) AS "expenses of yesterday" FROM website_log)

Query and analysis result

diff	\$
[6514393413.0,19578267596.0,0.33273594719539659]	

- The value 6514393413.0 indicates the expenses of yesterday.
- The value 19578267596.0 indicates the expenses of the same day last month.
- The value 0.33273594719539659 indicates the ratio of the expenses of yesterday to the expenses of the same day last month.

nullif() function

The nullif() function is used to check whether the values of two columns are the same. If the values are the same, null is returned. Otherwise, the value of expression1 is returned.

• Syntax

nullif(expression1, expression2)

- Example Check whether the values of the **client_ip** and **host** fields are the same.
- Query statement

* | SELECT nullif(client_ip,host)

· Query and analysis result

If the values of the client_ip and host fields are different, the value of the client_ip field is returned.

_col0	\$Q
61 198	
27	
111 - 52	
36	

try() function

The try() function is used to capture errors to ensure that Log Service can continue to query and analyze data.

- Syntax
- try(expression)
- Example

If an error occurs when the regexp_extract function is invoked, the try() function captures the error and Log Service continues to query and analyze data. The query and analysis result is returned.

Query statement

* | SELECT try(regexp_extract(request_uri, '.*\/(file.*)', 1)) AS file, count(*) AS count GROUP BY file

Query and analysis result

file 🗘 🗘	count \$
file-5	851
file-7	928
file-3	837
file-4	863

4.4.10.20. Nested subquery

This topic describes how to use nested subqueries when you query logs.

You can use nested gueries to perform more complicated gueries.

You must specify a FROM clause in the SQL statement of each nested query. However, this rule does not apply to non-nested queries. You must specify the from log keyword in each SQL statement to read raw data from logs.

```
Example:
* | select sum(pv) from
 (
 select count(1) as pv from log group by method
)
```

4.4.10.21. Array functions and operators

This topic describes the syntax of array functions and operators. This topic also provides examples on how to use the functions and operators. The following table describes the array functions and operators that are supported by Log Service.

Important If you want to use strings in analytic statements, you must enclose the strings in single quotation marks ("). Strings that are not enclosed or are enclosed in double quotation marks ("") indicate field names or column names. For example, 'status' indicates the status string, and status or "status" indicates the status log field.

Cloud Defined Storage

Function	Syntax	Description
Subscript operator	[x]	Returns the element whose index is <i>x</i> from an array. This operator is equivalent to the element_at function.
array_agg function	array_agg(x)	Returns an array that consists of all values inx.
array_distinct function	array_distinct(x)	Removes duplicate elements from an array.
array_except function	array_except(<i>x</i> , <i>y</i>)	Returns the difference between two arrays.
array_intersect function	array_intersect(x, y)	Returns the intersection of two arrays.
	array_join(<i>x, delimiter</i>)	Concatenates the elements in an array into a string by using a specified delimiter. If the array contains a null element, the null element is ignored. ① Important The array_join function can return a maximum of 1 KB of data. If the size of the returned data exceeds 1 KB, the excess data is truncated.
array_join function	array_join(<i>x, delimiter, null_replacement</i>)	Concatenates the elements in an array into a string by using a specified delimiter. If the array contains a null element, the null element is replaced by the value of the <i>null_replacement</i> parameter. ① Important The array_join function can return a maximum of 1 KB of data. If the size of the returned data exceeds 1 KB, the excess data is truncated.
array_max function	array_max(<i>x</i>)	Returns the maximum value in an array.
array_min function	array_min(<i>x</i>)	Returns the minimum value in an array.
array_position function	array_position(<i>x, element</i>)	Returns the index of a specified element in an array. The index starts from 1. If the specified element does not exist, the function returns 0.
array_remove function	array_remove(<i>x, element</i>)	Removes a specified element from an array.
array_sort function	array_sort(<i>x</i>)	Sorts the elements in an array in ascending order. If the array contains a null element, the null element is placed at the end.
array_transpose function	array_transpose(<i>x</i>)	Transposes a matrix and returns a new two-dimensional array that consists of the elements in the matrix. The elements are located by using the same indexes.
array_union function	array_union(x, y)	Returns the union of two arrays.
cardinality function	cardinality(x)	Returns the number of elements in an array.
concat function	concat(<i>x</i> , <i>y</i>)	Concatenates multiple arrays into one array.
contains function	contains(<i>x, element</i>)	Checks whether an array contains a specified element. If the array contains the specified element, the function returns true.
element_at function	element_at(<i>x</i> , <i>y</i>)	Returns the element whose index is <i>y</i> from an array.
filter function	filter(<i>x</i> , <i>lambda_expression</i>)	Filters elements in an array based on a lambda expression and returns elements that match the lambda expression.
flatten function	flatten(<i>x</i>)	Transforms a two-dimensional array into a one-dimensional array.
reduce function	reduce(<i>x</i> , <i>lambda_expression</i>)	Returns the sum of the elements in an array based on a lambda expression.
reverse function	reverse(<i>x</i>)	Reverses the elements in an array.
sequence function	sequence(<i>x</i> , <i>y</i>)	Returns an array of elements within a specified range. The elements are consecutive and incremental. The default incremental step is 1.
	sequence(<i>x, y, step</i>)	Returns an array of elements within a specified range. The elements are consecutive and incremental. The incremental step is a custom value.
shuffle function	shuffle(<i>x</i>)	Shuffles the elements in an array.
slice function	slice(<i>x, start, length</i>)	Returns a subset of an array.
transform function	transform(<i>x</i> , <i>lambda_expression</i>)	Transforms each element in an array by using a lambda expression.
zip function	zip(<i>x</i> , <i>y</i>)	Merges multiple arrays into a two-dimensional array. Elements that have the same index in the input arrays form a new array in the two-dimensional array.
zip_with function	<pre>zip_with(x, y, lambda_expression)</pre>	Merges two arrays into a single array by using a lambda expression.

Subscript operator

The subscript operator is used to return the element whose index is *x* from an array. This operator is equivalent to the element_at function.

	Suptox		
•			
	Parameters		
•	Persenter	Providelar.	
	Parameter		
	X	The index of an element in an array. The index starts from 1. The value of this parameter is of the bigint type.	
•	Return value type		
	The data type of the specified element.		
•	Example Obtain the first element from the value of the num	nher field	
	 Sample field 		
	number:[49,50,45,47,50]		
	Query statement		
	* SELECT cast(json_parse(number) as array(h	bigint)) [1]	
	 Query and analysis result 		
	co10	A 0	
	_000		
	49		
a	rray_agg function		
Th	ne array_agg function is used to return an array that	at consists of all values in x.	
•	Syntax		
array_agg (x)			
•	Parameters		
	Parameter	Description	
	x	The value of this parameter is of an arbitrary data type.	
•	Return value type		
	The array type.		
•	Example		
	Obtain an array that consists of all values in the st	atus field.	
4	Query statement		
	* SELECT array_agg(status) AS array		
4	 Query and analysis result 		
	array	:	
	[200,200,200,200,200,200,200,200,200,200		
a	rray_distinct function		
Th	ne array_distinct function is used to remove duplica	ate elements from an array.	
•	Syntax		
	array_distinct(x)		
Parameters			
	Parameter	Description	
	X	The value of this parameter is of the array type.	
•	Return value type		
	The array type.		

• Example

Remove duplicate elements from the value of the **number** field.

• Sample field

number:[49,50,45,47,50]

• Query statement

*| SELECT array_distinct(cast(json_parse(number) as array(bigint)))

• Query and analysis result

_col0	\$ Q,
[49,50,45,47]	

array_except function

The array_except function is used to return the difference between two arrays.

• Syntax

array_except(x, y)

• Parameters

Parameter	Description
x	The value of this parameter is of the array type.
У	The value of this parameter is of the array type.

Return value type

The array type.

• Example

Obtain the difference between the [1,2,3,4,5] and [1,3,5,7] arrays.

Query statement

* | SELECT array_except(array[1,2,3,4,5],array[1,3,5,7])

0	Query and analysis result	
	_col0	\$ Q
	[2,4]	

array_intersect function

The array_intersect function is used to return the intersection of two arrays.

Syntax

array_intersect(x, y)

• Parameters

Parameter	Description
X	The value of this parameter is of the array type.
У	The value of this parameter is of the array type.

Return value type

The array type.

- Example
 - Obtain the intersection of the [1,2,3,4,5] and [1,3,5,7] arrays.
- Query statement

*	L	SELECT	array	intersect	(array[1,	2,3,	4,5]	,array[1,3,5	5,7])
---	---	--------	-------	-----------	-----------	------	------	--------------	-------

Query and analysis result

_col0	\$ ٩
[1 3 5]	
[155]	

array_join function

The array_join function is used to concatenate the elements of an array into a string by using a specified delimiter.

Syntax

• The following syntax of the array_join function is used to concatenate the elements of an array into a string by using a specified delimiter. If the array contains a null element, the null element is ignored.

array_join(x, delimiter)

• The following syntax of the array_join function is used to concatenate the elements of an array into a string by using a specified delimiter. If the array contains a null element, the null element is replaced by the value of the null_replacement parameter.

array_join(x, delimiter,null_replacement)

• Parameters

Parameter	Description
x	The value of this parameter is of an arbitrary array type.
delimiter	The delimiter that is used to connect elements. You can specify a string for this parameter.
null_replacement	The string that is used to replace a null element.

- Return value type
 The varchar type.
- Example

Concatenate the elements of the [null, 'Log','Service'] array into a string by using space characters and replace the null element with Alicloud.

Query statement



array_max function

The array_max function is used to return the maximum value in an array.

- Syntax
- array_max(x)
- Parameters

andirectis				
Parameter	Description			
x	The value of this parameter is of the array type. Important If an array contains a null element, the function returns null.			

Return value type

The data type of elements in the parameter value.

• Example

O	tain the maximum value in an array.						
0	• Sample field						
	number: [49,50,45,47,50]						
Query statement							
	* SELECT array_max(try_cast(json_parse(number) as array(bigint))) AS max_number						
0	Query and analysis result						
	max_number \$\alpha\						
	50						

array_min function

The array_min function is used to return the minimum value in an array.

- Syntax
 - array_min(x)
- Parameters

Parameter	Description
X	The value of this parameter is of the array type. ① Important If an array contains a null element, the function returns null.

Return value type

The data type of elements in the parameter value.

• Example

0

Obtain the minimum value in an array.

- Sumple new

number:[49,50,45,47,50]
Query statement
* SELECT array_min(try_cast(json_parse(number) as array(bigint))) AS min_number
Query and analysis result
min_number \$Q
45

array_position function

The array_position function is used to return the index of a specified element in an array. The index starts from 1. If the specified element does not exist, the function returns 0.

• Syntax

array_position(x, element)

• Parameters

Parameter	Description
x	The value of this parameter is of the array type.
element	The value of this parameter is the element whose index you want to obtain. ③ Note If the element is null, the function returns null.
Return value type	

- The bigint type.
- Example
 - Obtain the index of 45 from the [49,45,47] array.
 - Query statement

* | SELECT array_position(array[49,45,47],45)

• Query and analysis result

_col0	\$ Q.
2	<u> </u>

array_remove function

The array_remove function is used to remove a specified element from an array.

• Syntax

array_remove(x, element)

• Parameters

Parameter	Description	
x	The value of this parameter is of the array type.	
element	The value of this parameter is the element that you want to remove. ⑦ Note If the element is null, the function returns null.	

- Return value type
- The array type.
- Example

Remove 45 from the [49,45,47] array.

• Query statement

* | SELECT array_remove(array[49,45,47],45)

 $\circ~$ Query and analysis result

_col0	\$ Q
[49.47]	
[,]	

array_sort function

The array_sort function is used to sort the elements in an array in ascending order. If the array contains a null element, the null element is placed at the end.

 Syntax 	
----------------------------	--

array_sort(x) • Parameters Parameter Description The value of this parameter is of the array type. х Return value type The array type. • Example Sort the elements in the ['b', 'd', null, 'c', 'a'] array in ascending order. • Query statement * | SELECT array_sort(array['b','d',null,'c','a']) • Query and analysis result _col0 ["a","b","c","d",null]

array_transpose function

The array_transpose function is used to transpose a matrix and return a new two-dimensional array that consists of the elements in the matrix. The

elements are located by using the same indexes. • Syntax		
array transpose(x)		
irameters		
Parameter	Description	
Parameter		
X	The value of this parameter is of the array(double) type.	
 Return value type The array(double) type. Example Create a two-dimensional array from elements that are located by using the same indexes in a different two-dimensional array. For example, in the [0,1,2,3], [10,19,18,17], and [0,9,8,7] arrays, 0, 10, and 9 are all located by using the index 1. This way, the new array [0.0,10.0,9.0] is formed. Query statement 		
<pre>* SELECT array_transpose(array[array[0,1,2])</pre>	,3],array[10,19,18,17],array[9,8,7]])	
Query and analysis result		
_col0	\$ Q.	
[[0.0, 10.0, 9.0], [1.0, 19.0, 8.0], [2.0, 18.0, 7.0], [3.0, 17.0]]		
array union function		
The array_union function is used to return the union	of two arrays.	
• Syntax		
array_union(x, y)		
Parameters		
Parameter	Description	
x	The value of this parameter is of the array type.	
V	The value of this parameter is of the array type	
 Return value type The array type. Example Obtain the union of the [1,2,3,4,5] and [1,3,5,7] arrays. 		
* SELECT array_union(array[1,2,3,4,5],array	y[1,3,5,7])	
 Query and analysis result 		
_col0 \$ Query and analysis result		
cardinality function		
The cardinality function is used to return the number of elements in an array.		
cardinality(x)		
Parameters		
Parameter	Description	
x	The value of this parameter is of the array type.	
 Return value type The bigint type. Example Obtain the number of elements in the value of the Sample field 	number field.	
number:[49,50,45,47,50]		

- Query statement

*| SELECT cardinality(cast(json_parse(number) as array(bigint)))

• Query and analysis result

_col0	\$ Q,
5	

concat function

concat(x, y...)

The concat function is used to concatenate multiple arrays into one array.

Syntax	
--------	--

Parameters

Parameter	Description	
x	The value of this parameter is of the array type.	
У	The value of this parameter is of the array type.	

Return value type

The array type.

• Example

Concatenate the ['red', 'blue'] and ['yellow', 'green'] arrays into one array.

Query statement

* SELECT con	<pre>ncat(array['red', 'blue'</pre>],array['yellow'	,'green'])
----------------	-------------------------------------	------------------	------------

0	Query and analysis result		
	_col0	\$ Q	
	["red","blue","yellow","green"]		

contains function

The contains function is used to check whether an array contains a specified element. If the array contains the specified element, the function returns true.

• Syntax

contains(x, element)

• Parameters

Parameter	Description
x	The value of this parameter is of the array type.
element	The value of this parameter is the element that you want to check.

Return value type

The Boolean type.

• Example

Check whether the value of the **region** field contains cn-beijing.

Sample field

region:["cn-hangzhou","cn-shanghai","cn-beijing"]

• Query statement

*| SELECT contains(cast(json_parse(region) as array(varchar)),'cn-beijing')

• Query and analysis result

col0	÷	C
rue		4

element_at function

The element_at function is used to return the element whose index is *y* from an array.

- Syntax
- element_at(x, y)
- Parameters

Parameter	Description
x	The value of this parameter is of the array type.
У	The index of an element in an array. The index starts from 1. The value of this parameter is of the bigint type.

Return value type

An arbitrary data type.

• Example

Obtain the second element from the value of the **number** field.

• Sample field			
	number:[49,50,45,47,50]		
0	Query statement		
	<pre>* SELECT element_at(cast(json_parse(number) AS array(varchar)), 2)</pre>		
0	Query and analysis result		
	_col0 🗘		
	50		

filter function

The filter function is used to filter elements in an array based on a lambda expression and return elements that match the lambda expression.

Syntax

filter(x, lambda_expression)

• Parameters

Parameter	Description
x	The value of this parameter is of the array type.
lambda_expression	The lambda expression. For more information, seeLambda expressions.

- Return value type
- The array type.
- Example

Obtain the elements that are greater than 0 from the [5,-6,null,7] array by using the lambda expression $x \rightarrow x > 0$.

• Query statement

	<pre>* SELECT filter(array[5,-6,null,7],x -> x > 0)</pre>	
0	Query and analysis result	
	_col0	÷ 0,
	[5.7]	

flatten function

The flatten function is used to transform a two-dimensional array into a one-dimensional array.

- Syntax
- flatten(x)

Parameters	
Parameter	Description
x	The value of this parameter is of the array type.

- Return value type
- The array type.
- Example
- Transform the two-dimensional array [array[1,2,3,4], array[5,2,2,4] into a one-dimensional array.
- Query statement

* | SELECT flatten(array[array[1,2,3,4],array[5,2,2,4]])

• Query and analysis result

_col0
[1,2,3,4,5,2,2,4]

reduce function

The reduce function is used to return the sum of the elements in an array based on a lambda expression.

Syntax

•

reduce(x, lambda_expression)		
Parameters		
Parameter	Description	
x	The value of this parameter is of the array type.	

Cloud Defined Storage

lambda_expression	The lambda expression. For more information, seeLambda expressions.	
Return value type		
The bigint type.		
• Example		
Obtain the sum of the elements in the [5,20,50] array.		
Query statement		
* SELECT reduce(array[5,20,50],0,(s, x	\rightarrow s + x, s \rightarrow s)	
Query and analysis result		
_col0	\$ Q.	
75	A	
reverse function		
The reverse function is used to reverse the elen	ients in an array.	
• Syntax		
reverse(x)		
Parameters		
Parameter	Description	
x	The value of this parameter is of the array type.	
Return value type		
The array type.		
• Example		
Reverse the elements in the [1,2,3,4,5] array.		
Query statement		
* SELECT reverse(array[1,2,3,4,5])		
 Query and analysis result 		
_col0	\$ Q.	
[5,4,3,2,1]	A	
sequence function		
The sequence function is used to return an arra	y of elements within a specified range. The elements are consecutive and incremental.	
• Syntax	,	
 The following syntax of the sequence function uses the default incremental step. The default incremental step is 1. 		
sequence(x, y)		
• The following syntax of the sequence functi	on uses a custom incremental step:	
sequence(x, y, step)		
Parameters		
Parameter	Description	
x	The value of this parameter is of the bigint or timestamp type. UNIX timestamps and date and time expressions are supported.	
у	The value of this parameter is of the bigint or timestamp type. UNIX timestamps and date and time expressions are supported.	

TI If OI	e incremental step. the values of the x and y parameters are date and time expressions, the value of the tep parameter is in le of the following formats:
0	interval ' n' year to month : The incremental step is n years.
0	interval 'n' day to second : The incremental step is n days.

- Return value type
- The array type.
- Examples

step

User Guide-Log Service

c	 Example 1: Obtain the even numbers within the range from 0 to 10. Query statement 		
	* SELECT sequence(0,10,2)		
	Query and analysis result		
	_col0	¢ م.	
	[0,2,4,6,8,10]		
c	Example 2: Obtain the dates within the range fr Query statement	om 2017-10-23 to 2021-08-12 at the incremental step of 1 year.	
	ww* SELECT sequence(from_unixtime(15087	37026),from_unixtime(1628734085),interval '1' year to month)	
	Query and analysis result		
	["2017-10-23 13:37:06.000","2018-10-23 13:37:06.000"," 00"]	2019-10-23 13:37:06.000","2020-10-23 13:37:01	
c	Example 3: Obtain the UNIX timestamps within Query statement 	the range from 1628733298 to 1628734085 at the incremental step of 60 seconds.	
	* SELECT sequence(1628733298,1628734085	,60)	
	Query and analysis result		
	_col0	\$ Q.	
	[1628733298,1628733358,1628733418,1628733478,1628 28733778,1628733838,1628733898,1628733958,162873	733538,1628733598,1628733658,1628733718. 4018,1628734078] Hide	
sł Th	uffle function e shuffle function is used to shuffle the elements iyntax	in an array.	
	<pre>shuffle(x)</pre>		
Parameters			
	Parameter	Description	
	x	The value of this parameter is of the array type.	
Return value type			
-	he array type.		
•	wample		
	Query statement		
	<pre>* SELECT shuffle(array[1,2,3,4,5])</pre>		
c	Query and analysis result		
	_col0	¢ Q.	
	[3,1,2,4,5]		
[5,1,2,4,3]			
	[2,5,3,1,4]		
sl Th	ice function a slice function is used to return a subset of an ar	ray.	
• 9	iyntax		
slice(x, start, length)			
Parameters			
	Parameter	Description	
	X	The value of this parameter is of the array type.	
	start	 The index at which Log Service starts to extract elements. If the value of the <i>start</i> parameter is negative, Log Service starts to extract elements from the end of the array. If the value of the <i>start</i> parameter is a positive number, Log Service starts to extract elements from the beginning of the array. 	

The number of elements that you want to include in the subset.

Return value type

length

The array type.

- Example
 - Obtain a subset of the [1,2,4,5,6,7,7] array from the third element with two elements.
 - Query statement

* SELECT slice(array[1,2,4,5,6,7,7],3	,2)
---	-----

Query and analysis result

_col0	\$ Q.
[4,5]	A

transform function

The transform function is used to transform each element in an array by using a lambda expression.

Syntax

transform(x, lambda_expression)

• Parameters

Parameter	Description
x	The value of this parameter is of the array type.
lambda_expression	The lambda expression. For more information, seeLambda expressions.

- Return value type
- The array type.
- Example

Add 1 to each element in the [5,6] array and return a new array.

Query statement

* | SELECT transform(array[5,6],x -> x + 1)

0	Query and analysis result	
	_col0	: Q
	[6.7]	

zip function

The zip function is used to merge multiple arrays into a two-dimensional array. Elements that have the same index in the input arrays form a new array in the two-dimensional array.

- Syntax
 - zip(x, y...)
- Parameters

Parameter	Description
x	The value of this parameter is of the array type.
У	The value of this parameter is of the array type.

- Return value type
- The array type.
- Example

Merge the [1, 2,3], ['1b', null, '3b'], and [1, 2,3] arrays into a two-dimensional array.

Query statement

* | SELECT zip(array[1,2,3], array['1b',null,'3b'],array[1,2,3])

0	Query	and	analysis	result	
---	-------	-----	----------	--------	--

_col0	\$ Q.
[[1,"1b",1],[2,null,2],[3,"3b",3]]	A

zip_with function

The zip_with function is used to merge two arrays into a single array by using a lambda expression.

Syntax

zip_with(x, y, lambda_expression)

Parameters

Parameter	Description
x	The value of this parameter is of the array type.
/	The value of this parameter is of the array type.
User Guide-Log Service

•

•

lambda_expression	The lambda expression. For more information, seeLambda expressions.
Return value type	
The array type.	
Example	
Use the lambda expression $({\rm x},~{\rm y}) ~\rightarrow~ {\rm x} + {\rm y}$ to a	add the elements in the [1, 2] and [3, 4] arrays and return a new array.
Query statement	
<pre>SELECT zip_with(array[1,2], array[3,4],(x,y)</pre>	-> x + y)
 Query and analysis result 	
_col0	\$ 0,
[4,6]	A

4.4.10.22. Binary string functions

This topic describes the syntax of binary string functions. This topic also provides examples on how to use the functions. Varbinary data is different from varchar data.

Function	Description
Concatenation operator ()	The result of a b is ab .
length(binary)	Returns the length of a binary string in bytes. The return value is of the bigint type.
concat(binary1,, binaryN)	Concatenates binary strings. This function is equivalent to $.$ The return value is of the varbinary type.
to_base64(binary)	Converts a binary string to a Base64 string. The return value is of the varchar type.
from_base64(string)	Converts a Base64 string to a binary string. The return value is of the varbinary type.
to_base64url(binary)	Converts a string to a URL-safe Base64 string. The return value is of the varchar type.
from_base64url(string)	Converts a URL-safe Base64 string to a binary string. The return value is of the varbinary type.
to_hex(binary)	Converts a binary string to a hexadecimal string. The return value is of the varchar type.
from_hex(string)	Converts a hexadecimal string to a binary string. The return value is of the varbinary type.
to_big_endian_64(bigint)	Converts a number to a binary string in big endian mode. The return value is of the varbinary type.
from_big_endian_64(binary)	Converts a binary string in big endian mode to a number. The return value is of the bigint type.
md5(binary)	Calculates the MD5 value of a binary string. The return value is of the varbinary type.
shal(binary)	Calculates the SHA1 value of a binary string. The return value is of the varbinary type.
sha256(binary)	Calculates the SHA256 hash value of a binary string. The return value is of the varbinary type.
sha512(binary)	Calculate the SHA512 value of a binary string. The return value is of the varbinary type.
xxhash64(binary)	Calculates the xxhash64 value of a binary string. The return value is of the varbinary type.

4.4.10.23. Bitwise functions

This topic describes the syntax of bitwise functions. This topic also provides examples on how to use the functions.

Function	Description	Example		
bit_count(x, bits)	Counts the number of 1s in x in two's complement. The x variable is a signed integer that includes the specified number of bits. The return value is of the bigint type.	 SELECT bit_count (9, 64) returns 2. SELECT bit_count (9, 8) returns 2. SELECT bit_count (-7, 64) returns 62. SELECT bit_count (-7, 8) returns 6. 		
bitwise_and(x, y)	Returns the bitwise AND of ${\sf x}$ and ${\sf y}$ in two's complement. The return value is of the bigint type.	None		
bitwise_not(x)	Returns the bitwise NOT of x in two's complement. The return value is of the bigint type.	None		

bitwise_or(x, y)	Returns the bitwise OR of x and y in two's complement. The return value is of the bigint type.	None
bitwise_xor(x, y)	Returns the bitwise XOR of \boldsymbol{x} and \boldsymbol{y} in two's complement. The return value is of the bigint type.	None

4.4.10.24. Interval-valued comparison and periodicity-valued comparison

functions

Log Service supports interval-valued comparison and periodicity-valued comparison functions. You can use these functions to query and analyze log data.

compare() function

The compare() function is used to compare the calculation result of the current time period with the calculation result of a time period N seconds before. You can use this function to perform an interval-valued comparison or periodicity-valued comparison on data.

Syntax

compare(column name,N)			
⑦ Note The compare() function of me,N1,N2,N3).	an be used to compare the calcula	tion results of multiple periods of	time, for example, compare(<i>column na</i>
column name: the name of the specif N: the time window. Unit: seconds. Ex	ied column. The value of this parar ample: 3600 (1 hour), 86400 (one	neter must be of the double type day), or 604800 (one week).	or long type.
Response			
The returned result is a JSON array in the for the seconds before, the UNIX timestam	e following format: [the current va p before N seconds]. Example: [11	lue, the value before N seconds, t 76.0,1180.0,0.996610169491525	he ratio of the current value to the value 5,1611504000.0].
Examples Calculate the ratio of the page views	(PVs) of the current hour to the PV	s of the same time period the day	before.
Set the time range to 1 Hour(Time F (one day). log indicates the Logstore	rame) and execute the following on name.	query statement. 86400 indicates	the current time minus 86400 seconds
* SELECT compare(PV, 86400) FROM	(SELECT count(*) AS PV FROM log)		
The following figure shows the return	ed result.		
_col0			\$ 0.
[3337.0,3522.0,0.947473026689381]			
 3337.0 indicates the PVs of the cur 3522.0 indicates the PVs of the sar 0.947473026689381 indicates the To display the analysis result in multiplates the same set of the	rent 1 hour, for example, Dec 25, 2 ne time period the day before, for e ratio of the PVs of the current ho ple columns, you can execute the f	2020, 14:00:00 ~ Dec 25, 2020, 1 example, Dec 24, 2020, 14:00:00 ur to the PVs of the same time per following query statement:	5:00:00. ~ Dec 24, 2020, 15:00:00. iod the day before.
<pre>* SELECT diff[1] AS today, diff[2 FROM log))</pre>	2] AS yesterday, diff[3] AS ratio	FROM (SELECT compare(PV,86400) A	S diff FROM (SELECT count(*) AS PV
The following figure shows the return	ed result.		
today 💠 🔾	yesterday	¢ ⊂ ratio	\$ Q
.3337.0	.3522.0	0.947473026689381	

? Note

To compare the data of a specified year or week with the data of the previous year or week, you can use the query statements in the following examples:

For example, if you want to calculate the ratio of the PVs of November 2020 to the PVs of November 2019, you can set the time range to Nov 1, 2020, 00:00~Dec 1, 2020, 00:00, and execute the following query statement:

* | SELECT compare(PV, 31622400) FROM (SELECT count(*) AS PV FROM log)

 For example, if you want to calculate the ratio of the PVs of a Tuesday to the PVs of the previous Tuesday, you can set the time range to Jan 18, 2021, 00:00~Jan 19, 2021, 00:00, and execute the following query statement:

* | SELECT compare(PV, 604800) FROM (SELECT count(*) AS PV FROM log)

Calculate the ratio of the PVs of each hour of the current day to the PVs of the same time period the day before and two days before.
 Set the time range to Today(Time Frame) and execute the following query statement. 86400 indicates the current time minus 86400 seconds (one day). 172800 indicates the current time minus 172800 seconds (two days). log indicates the Logstore name.
 date_format(from_unixtime(_time_), '%H:00') indicates the format of the returned time.

* | SELECT time, compare(PV, 86400,172800) as diff from (SELECT count(*) as PV, date_format(from_unixtime(__time__), '%H:00') as time from log GROUP BY time) GROUP BY time ORDER BY time

The following figure shows the returned result.

time \$ Q	diff \$\\$ Q
2022-09-22 00:00:00.000	[1174.0,1191.0,1253.0,0.9857262804366079,0.93695131683958 5,1663689600.0,1663603200.0]
2022-09-22 01:00:00.000	[9765.0,9930.0,10120.0,0.9833836858006042,0.9649209486166 008,1663693200.0,1663606800.0]
2022-09-22 02:00:00.000	[27649.0,28146.0,27314.0,0.9823420734740282,1.01226477264 40653,1663696800.0,1663610400.0]
2022-09-22 03:00:00.000	[35485.0,37092.0,35602.0,0.9566752938639059,0.99671366777 14735,1663700400.0,1663614000.0]
2022-09-22 04:00:00.000	[27097.0,27370.0,26849.0,0.9900255754475703,1.00923684308 54036,1663704000.0,1663617600.0]

• 1176.0 indicates the PVs of the current time period, for example, Dec 25, 2020, 00:00 ~ Dec 25, 2020, 01:00.

• **1180** indicates the PVs of the same time period the day before, for example, Dec 24, 2020, 00:00 ~ Dec 24, 2020, 01:00.

• 1167.0 indicates the PVs of the same time period two days before, for example, Dec 23, 2020, 00:00:00 ~ Dec 23, 2020, 01:00:00.

• 0.9966101694915255 indicates the ratio of the PVs of the current time period to the PVs of the same time period the day before.

• 1.0077120822622108 indicates the ratio of the PVs of the current period to the PVs of the same period two days before.

To display the analysis result in multiple columns, you can execute the following query statement:

* | SELECT time, diff[1] AS day1, diff[2] AS day2, diff[3] AS day3, diff[4] AS ratio1, diff[5] AS ratio2 FROM (SELECT time, compare(PV, 86 400,172800) as diff from (SELECT count(*) as PV, date_format(from_unixtime(__time__), '%H:00') as time from log GROUP BY time) GROUP BY time ORDER BY time)

The fo	llowing	figure	shows	the re	eturned	resul	t.

time	• Q	day1 🗘 Q	day2 🗘 Q	day3 \$ Q	ratio1 \$ Q	ratio2 🗘 Q
2022-09-22 00:00:00.000		1174	1191	1253	0.98572628	0.93695132
2022-09-22 01:00:00.000		9765	9930	10120	0.98338369	0.96492095
2022-09-22 02:00:00.000		27649	28146	27314	0.98234207	1.01226477
2022-09-22 03:00:00.000		35485	37092	35602	0.95667529	0.99671367

• Calculate the ratio of the PVs of December to the PVs of November in the same year.

Set the time range to **This Month(Time Frame)** and execute the following query statement. **2592000** indicates the current time minus 2592000 seconds (one month). **Iog** indicates the Logstore name. **date_trunc('month', __time__)** indicates that the date_trunc function is used to truncate a point in time by month.

| SELECT time, compare(PV, 2592000) AS diff from (SELECT count() AS PV, date_trunc('month', __time__) AS time from log GROUP BY time) GR OUP BY time ORDER BY time

The following figure shows the returned result.

time 🗘 🗘	diff \$\$ 0,	
2021-01-01 00:00:00.000	[11958378.0,448571.0,26.658829928818404]	

ts_compare() function

ts compare(column name,N)

The ts_compare() function is used to compare the calculation result of the current time period with the calculation result of a time period N seconds before. You can use this function to perform an interval-valued comparison or periodicity-valued comparison on data. The analysis results of the ts_compare() function must be grouped by the time column by using GROUP BY clauses.

Syntax

Note The ts_compare() function can be used to compare the calculation results of multiple periods of time, for example, ts_compare(*column name*,N1,*N2*,*N3*).

• column name: the name of the specified column. The value of this parameter must be of the double type or long type.

• N: the time window, Unit: seconds, Example: 3600 (1 hour), 86400 (one day), or 604800 (one week),

Response

The returned result is a JSON array in the following format: [the current value, the value before N seconds, the ratio of the current value to the value of N seconds before, the UNIX timestamp before N seconds]. Example: [1176.0,1180.0,0.9966101694915255,1611504000.0].

Example

Calculate the ratio of the PVs of every hour today to the PVs of the previous hour.

Set the time range to **Today(Relative)** and execute the following query statement. **3600** indicates the current time minus 3600 seconds (1 hour). **log** indicates the Logstore name. **date_trunc('hour',_time_)** indicates that the date_trunc function is used to truncate the time by hour.

* | SELECT time, ts_compare(PV, 3600) AS data FROM(SELECT date_trunc('hour',__time__) AS time, count(*) AS PV from log GROUP BY time ORDER BY time) GROUP BY time

The following figure shows the returned result		
time \$ Q	data	\$Q.
2021-01-27 00:00:00.000	[1160.0,10034.0,0.11560693641618497,1611673200.0]	•
2021-01-27 01:00:00.000	[10177.0,1160.0,8.773275862068966,1611676800.0]	
2021-01-27 02:00:00.000	[26804.0,10177.0,2.6337820575808195,1611680400.0]	

4.4.10.25. Comparison functions and operators

This topic describes the comparison functions and operators in Log Service. You can use these functions and operators to query and analyze log data. A comparison function compares the values of two parameters. The values can be one of the arbitrary comparable data types, such as integer, bigint, double, and text.

Comparison operators

A comparison operator is used to compare two values. If the statement is true, TRUE is returned. Otherwise, FALSE is returned.

Operator	Description
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
=	Equal to
<>	Not equal to
!=	Not equal to

Range operator BETWEEN

The BETWEEN operator is used to check whether a value falls in a specified closed interval.

If the value falls in the specified closed interval, TRUE is returned. Otherwise, FALSE is returned.
 Example: SELECT 3 BETWEEN 2 AND 6; The statement is true, and TRUE is returned.
 The preceding statement is equivalent to SELECT 3 >= 2 AND 3 <= 6;

The BETWEEN operator can be specified after the NOT operator to check whether a value falls out of a specified closed interval.
 Example: SELECT 3 NOT BETWEEN 2 AND 6; . The statement is false, and FALSE is returned.
 The preceding statement is equivalent to SELECT 3 < 2 OR 3 > 6; .

• If one of the three values is NULL, the result is NULL.

IS NULL and IS NOT NULL

The IS NULL and IS NOT NULL operators are used to check whether a value is NULL.

IS DISTINCT FROM and IS NOT DISTINCT FROM

The IS DISTINCT FROM and IS NOT DISTINCT FROM operators are similar to the EQUAL TO and NOT EQUAL TO operators. The difference is that the IS DISTINCT FROM and IS NOT DISTINCT FROM operators can be used to check whether a NULL value exists.

Examples:

```
SELECT NULL IS DISTINCT FROM NULL; -- false
SELECT NULL IS NOT DISTINCT FROM NULL; -- true
```

You can use the DISTINCT operator to compare parameter values under multiple conditions. The following table describes the conditions.

a	b	a = b	a <> b	a DISTINCT b	a NOT DISTINCT b
1	1	TRUE	FALSE	FALSE	TRUE
1	2	FALSE	TRUE	TRUE	FALSE
1	NULL	NULL	NULL	TRUE	FALSE
NULL	NULL	NULL	NULL	FALSE	TRUE

GREATEST and **LEAST**

The GREATEST operator is used to obtain the maximum value from multiple columns. The LEAST operator is used to obtain the minimum value from multiple columns.

Example:

select greatest(1,2,3) ; -- Returns 3.

Quantified comparison predicates: ALL, ANY, and SOME

The ALL, ANY, and SOME quantifiers are used to check whether a parameter value meets specified conditions.

- ALL is used to check whether a parameter value meets all conditions. If the statement is true, TRUE is returned. Otherwise, FALSE is returned.
- ANY is used to check whether a parameter value meets one of the specified conditions. If the statement is true, TRUE is returned. Otherwise, FALSE
- is returned.SOME is used to check whether a parameter value meets one of the specified conditions. SOME is equivalent to ANY.

• ALL, ANY, and SOME must be specified after comparison operators.

You can use ALL and ANY to compare values under multiple conditions. The following table describes the conditions.

Expression	Description
A = ALL ()	Returns TRUE if A matches all values.
A <> ALL ()	Returns TRUE if A does not match all values.
A < ALL ()	Returns TRUE if A is smaller than the smallest value.
A = ANY ()	Returns TRUE if A is equal to a value. This statement is equivalent to A IN ().
A <> ANY ()	Returns TRUE if A does not match a value.
A < ANY ()	Returns TRUE if A is smaller than the largest value.

Examples:

SELECT 'hello' = ANY (VALUES 'hello', 'world'); -- true

SELECT 21 < ALL (VALUES 19, 20, 21); -- false

SELECT 42 >= SOME (SELECT 41 UNION ALL SELECT 42 UNION ALL SELECT 43); -- true

4.4.10.26. Lambda expressions

Log Service allows you to define a lambda expression in an analytic statement and pass the expression to a specified function. This topic describes the syntax of lambda expressions. This topic also provides examples on how to use the expressions.

Syntax

You must use lambda expressions together with functions, such as filter function, reduce function, transform function, zip_with function, and map_filter function. Syntax:

parameter -> expression

Parameter	Description
parameter	The identifier that is used to pass parameters.
expression	The lambda expression, which can include most MySQL expressions. Examples: x -> x + 1 (x, y) -> x + y x -> regexp_like(x, 'a+') x -> x(1) / x(2] x -> if(x > 0, x, -x) x -> coalscc(x, 0) x -> cast(x AS JSON) x -> x + try(1 / 0)

Examples

- Example 1: x -> x is not null
- This lambda expression is used to return the non-null elements in the [5, null, 7, null] array.
- Query statement

* | SELECT filter(array[5, null, 7, null], x -> x is not null)

Query and analysis result

_col0	\$Q
[5,7]	

• Example 2: 0, (s, x) -> s + x, s -> s

This lambda expression is used to return the sum of the elements in the [5,20,50] array.

Query statement

* | SELECT reduce(array[5, 20, 50], 0, (s, x) -> s + x, s -> s)

• Query and analysis result

_col0	\$ Q
75	

[•] Example 3: (k,v) -> v > 10

This lambda expression is used to create a map from two arrays. The values of keys in the map are greater than 10.

Ouerv statement

* | SELECT map_filter(map(array['class01', 'class02', 'class03'], array[11, 10, 9]), (k,v) -> v > 10)

>	Query and analysis result	

_col0	\$ Q.	
{"class01":11}		

• Example 4: (x, y) -> (y, x)

This lambda expression is used to transpose the elements in two arrays and return a new two-dimensional array that is created from the elements in the two arrays. The elements are located by using the same indexes.

Query statement

0

* SELEC	<pre>T zip_with(array[1,</pre>	3, 5],	array['a',	'b',	'c'],	(x,	y)	->	(y,	x))	
Query and	analysis result										
_col0											\$ Q

[["a".1].["b".3].["c".5]]	[["a".1].["b".3].["c".5]]

• Example 5: x -> coalesce(x, 0) + 1

This lambda expression is used to add 1 to each element in the [5, NULL, 6] array and return the result. The null element in the array is converted to 0 before 1 is added.

• Query statement

* | SELECT transform(array[5, NULL, 6], x -> coalesce(x, 0) + 1)

0	Query	and	analysis	result
---	-------	-----	----------	--------

_col0	\$ Q
[6,7]	

Additional examples

- * | SELECT filter(array[], x -> true)
- * | SELECT map_filter(map(array[],array[]), (k, v) -> true)
- * | SELECT reduce(array[5, 6, 10, 20], -- calculates arithmetic average: 10.25 cast(row(0.0, 0) AS row(sum double, count integer)),
 - (s, x) -> cast(row(x + s.sum, s.count + 1) AS row(sum double, count integer)),
- $\label{eq:s-s} s \ -> \ if(s.count = 0, \ null, \ s.sum \ / \ s.count))$ * | SELECT reduce(array[2147483647, 1], cast(0 AS bigint), (s, x) -> s + x, s -> s)
- * | SELECT reduce(array[5, 20, null, 50], 0, (s, x) -> s + x, s -> s)
- * | SELECT transform(array[array[1, null, 2], array[3, null]], a -> filter(a, x -> x is not null))
- * | SELECT zip_with(array['a', 'b', 'c'], array['d', 'e', 'f'], (x, y) -> concat(x, y))

4.4.10.27. Logical functions

This topic describes the available logical functions in Log Service. You can use these functions to query and analyze log data.

Logical operators

Operator	Description	Example
AND	The result is TRUE if both values are TRUE.	a AND b
OR	The result is TRUE if either value is TRUE.	a OR b
NOT	The result is TRUE if the value is FALSE.	NOT a

Effect of NULL on logical operators

The following tables list the truth values when the values of a and b are TRUE, FALSE, and NULL, respectively. Table 1. Truth table 1

a	b	a AND b	a OR b
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
TRUE	NULL	NULL	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE
FALSE	NULL	FALSE	NULL
NULL	TRUE	NULL	TRUE
NULL	FALSE	FALSE	NULL
NULL	NULL	NULL	NULL

Table 2. Truth table 2

а	NOT a
TRUE	FALSE
FALSE	TRUE
NULL	NULL

4.4.10.28. Column aliases

This topic describes how to specify an alias for a column and provides some examples.

A column name in an SQL statement can contain only letters, digits, and underscores (_). The column name must start with a letter.

When you configure log collection, you may specify a column name that does not conform to the SQL standard, for example, User-Agent. In this case, you must specify an alias for the column in the Search & Analysis panel in which you can configure index attributes. The alias is used only if you execute an SQL statement to query and analyze logs. The original name of each column is stored. Therefore, you must search for columns by original name.

If the original name of a column is long, you can specify an alias for the column in an SQL statement. Table 1. Sample aliases

Original column name	Alias
User-Agent	ua
User.Agent	ua
123	col
abceefghijklmnopqrstuvw	a

4.4.10.29. JOIN queries on a Logstore and a MySQL database

Log Service allows you to use the JOIN syntax to query data from a Logstore and a MySQL database. The query results are saved to the database.

Prerequisites

An external store is created. For more information, see Associate Log Service with a MySQL database.

Background information

Procedure

- 1. Log on to the Log Service console.
- 2. In the Projects section, click the name of the project that you want to manage.
- 3. Choose Log Storage > Logstores. On the Logstores tab, click the Logstore that you want to manage.
- 4. Execute a query statement. Log Service supports the following JOIN syntax:
 - [INNER] JOIN LEFT [OUTER] JOIN

RIGHT [OUTER] JOIN FULL [OUTER] JOIN

The following sample code provides an example of a JOIN query. For more information, see JOIN clause.

method:postlogstorelogs | select count(1) , histogram(logstore) from log l join join_meta m on l.projectid = cast(m.ikey as varchar)

() Important

- You can use the JOIN syntax only on a Logstore and a small table in a MySQL database. A small table contains less than 20 MB of data.
 - In a query statement, the name of the Logstore must precede the join keyword, and the name of the external store must follow the join keyword.

• You must specify the name of the external store in a query statement. When the system executes the statement, the system replaces the name with the name of the database and the name of the table. Do not enter only the table name.

5. Save the query results to the MySQL database.

Log Service allows you to insert the query results into the database by using an INSERT statement. The following sample code provides an example of an INSERT statement:

method:postlogstorelogs | insert into method_output select cast(method as varchar(65535)),count(1) from log group by method

Sample Python script

encoding: utf-8 from __future__ import print_function from aliyun.log import * from aliyun.log.util import base64_encodestring from random import randint import time import os from datetime import datetime endpoint = os.environ.get('ALIYUN_LOG_SAMPLE_ENDPOINT', 'cn-chengdu.log.aliyuncs.com') accessKeyId = os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSID', '') accessKey = os.environ.get('ALIYUN_LOG_SAMPLE_ACCESSKEY', '')
logstore = os.environ.get('ALIYUN_LOG_SAMPLE_LOGSTORE', '') project = "ali-yunlei-chengdu" client = LogClient(endpoint, accessKeyId, accessKey, token) # Create an external store. res = client.create_external_store(project,ExternalStoreConfig("rds_store","region","rds-vpc","vpc id","Instance ID","Instance IP address", "Instance port", "Username", "Password", "Database name", "Table name")); res.log_print() # Retrieve the details of the external store. res = client.get_external_store(project,"rds_store"); res.log_print() res = client.list_external_store(project,""); res.log_print(); # Perform a JOIN query. req = GetLogsRequest(project,logstore,From,To,"","select count(1) from "+ logstore +" s join meta m on s.projectid = cast(m.ikey as varchar)"); res = client.get_logs(req) res.log_print(); # Save the query results to the MySQL database. req = GetLogsRequest(project,logstore,From,To,""," insert into rds_store select count(1) from "+ logstore); res = client.get_logs(req) res.log_print();

4.4.10.30. Geospatial functions

This topic describes the available geospatial functions in Log Service. You can use these functions to query and analyze log data.

Concept of geometry

Geospatial functions support geometries in the well-known text (WKT) format.

Table 1. Geometry formats

Geometry	WKT format
Point	POINT (0 0)
LineString	LINESTRING (0 0, 1 1, 1 2)
Polygon	POLYGON ((0 0, 4 0, 4 4, 0 4, 0 0), (1 1, 2 1, 2 2, 1 2, 1 1))
MultiPoint	MULTIPOINT (0 0, 1 2)
MultiLineString	MULTILINESTRING ((0 0, 1 1, 1 2), (2 3, 3 2, 5 4))
MultiPolygon	MULTIPOLYGON (((0 0, 4 0, 4 4, 0 4, 0 0), (1 1, 2 1, 2 2, 1 2, 1 1)), ((-1 -1, -1 -2,
GeometryCollection	GEOMETRYCOLLECTION (POINT(2 3), LINESTRING (2 3, 3 4))

Constructors

Table 2. Constructor description

Function	Description
$ST_Point(double, double) \rightarrow Point$	Returns a geometry point instance with the specified coordinate values.
$ST_LineFromText(varchar) \rightarrow LineString$	Returns a geometry LineString instance from a WKT representation.
$ST_Polygon(varchar) \rightarrow Polygon$	Returns a geometry polygon instance from a WKT representation.
$ST_GeometryFromText(varchar) \to Geometry$	Returns a geometry instance from a WKT representation.
$ST_AsText(Geometry) \rightarrow varchar$	Returns the WKT representation of a geometry.

Operations

Function	Description
$ST_Boundary(Geometry) \rightarrow Geometry$	Returns the closure of the combinatorial boundary of a geometry.
$ST_Buffer(Geometry, distance) \rightarrow Geometry$	Returns the geometry that represents all points whose distance from the specified geometry is shorter than or equal to the specified distance.
$ST_Difference(Geometry, Geometry) \rightarrow Geometry$	Returns the geometry value that represents the point set difference of the specified geometries.

$ST_Envelope(Geometry) \rightarrow Geometry$	Returns the bounding rectangular polygon of a geometry.
$ST_ExteriorRing(Geometry) \rightarrow Geometry$	Returns a line string that represents the exterior ring of the input polygon.
$ST_Intersection(Geometry,Geometry) \to Geometry$	Returns the geometry value that represents the point set intersection of two geometries.
ST_SymDifference(Geometry, Geometry) \rightarrow Geometry	Returns the geometry value that represents the point set symmetric difference of two geometries.

Relationship tests

Function	Description
ST_Contains(Geometry, Geometry) \rightarrow boolean	Returns True if and only if no points of the second geometry lie in the exterior of the first geometry, and at least one point of the interior of the first geometry lies in the interior of the second geometry. Returns False if points of the second geometry are on the boundary of the first geometry.
$ST_Crosses(Geometry, Geometry) \rightarrow boolean$	Returns True if the specified geometries share some, but not all, interior points in common.
ST_Disjoint(Geometry, Geometry) \rightarrow boolean	Returns True if the specified geometries do not spatially intersect.
ST_Equals(Geometry, Geometry) \rightarrow boolean	Returns True if the specified geometries represent the same geometry.
ST_Intersects(Geometry, Geometry) \rightarrow boolean	Returns True if the specified geometries spatially intersect in two dimensions.
$ST_Overlaps(Geometry, Geometry) \rightarrow boolean$	Returns True if the specified geometries share space in the same dimension, but are not completely contained by each other.
$ST_Relate(Geometry, Geometry) \rightarrow boolean$	Returns True if the first geometry is spatially related to the second geometry.
ST_Touches(Geometry, Geometry) → boolean	Returns True if the specified geometries have at least one point in common, but their interiors do not intersect.
ST_Within(Geometry, Geometry) \rightarrow boolean	Returns True if the first geometry is completely inside the second geometry. Returns False if the two geometries have points in common at the boundaries.

Accessors

Function	Description
$ST_Area(Geometry) \rightarrow double$	Returns the two-dimensional Euclidean area of a geometry.
ST_Centroid(Geometry) → Geometry	Returns the point value that is the mathematical centroid of a geometry.
$ST_CoordDim(Geometry) \rightarrow bigint$	Returns the coordinate dimension of a geometry.
$ST_Dimension(Geometry) \rightarrow bigint$	Returns the inherent dimension of a geometry object, which must be less than or equal to the coordinate dimension.
ST_Distance(Geometry, Geometry) \rightarrow double	Returns the minimum two-dimensional Cartesian distance (based on spatial ref) between two geometries in projected units.
$ST_lsClosed(Geometry) \rightarrow boolean$	Returns True if the start and end points of the linestring are coincident.
ST_IsEmpty(Geometry) → boolean	Returns True if the specified geometry is an empty geometry, such as geometry collection, polygon, and point.
$ST_IsRing(Geometry) \rightarrow boolean$	Returns True if and only if the line is closed and simple.
ST_Length(Geometry) → double	Returns the length of a LineString or multi-LineString by using Euclidean measurement on a two-dimensional plane (based on spatial ref) in projected units.
$ST_XMax(Geometry) \rightarrow double$	Returns the X maximum of the bounding box of the geometry.
$ST_YMax(Geometry) \rightarrow double$	Returns the Y maximum of the bounding box of the geometry.
$T_XMin(Geometry) \rightarrow double$	Returns the X minimum of the bounding box of the geometry.
$ST_YMin(Geometry) \rightarrow double$	Returns the Y minimum of the bounding box of the geometry.
$ST_StartPoint(Geometry) \rightarrow point$	Returns the first point of a geometry LineString instance.
$ST_EndPoint(Geometry) \rightarrow point$	Returns the last point of a geometry LineString instance.
$ST_X(Point) \rightarrow double$	Returns the X coordinate of a point.
$ST_Y(Point) \rightarrow double$	Returns the Y coordinate of a point.
$ST_NumPoints(Geometry) \rightarrow bigint$	Returns the number of points in a geometry.
$ST_NumInteriorRing(Geometry) \rightarrow bigint$	Returns the cardinality of the collection of interior rings of a polygon.

4.4.10.31. Geography functions

This topic describes the syntax of geography functions and provides some examples.

IP functions identify the country, province, city, Internet service provider (ISP), and longitude and latitude of a specific IP address. For more information, see IP functions.

Function	Description	Example

geohash(string)	Returns the geohash value of a specified geographical coordinate. The geographical coordinate is represented by a string in the format of " <latitude>, <longitude>". The return value is a string. Example: geohash('34.1,120.6').</longitude></latitude>	<pre>* select geohash('34.1,120.6') = 'wwjcbrdnzs'</pre>
geohash(lat,lon)	Returns the geohash value of a specified geographical coordinate. The geographical coordinate is represented by two separate parameters that indicate the latitude and longitude. The return value is a string.	<pre>* select geohash(34.1,120.6) = 'wwjcbrdnzs'</pre>

4.4.10.32. JOIN clause

You can use JOIN clauses in SQL statements to join multiple tables by using fields that are shared by the tables. In Log Service, you can join one or more Logstores. You can also join Logstores with ApsaraDB RDS instances. This topic describes how to join different Logstores.

Procedure

- 1. Download the latest version of the Log Service SDK for Python.
- 2. Call the GetProjectLogs operation to query logs.

Sample SDK

#!/usr/bin/env python
#encoding: utf-8
import time,sys,os
from aliyun.log.logexception import LogException
from aliyun.log.logitem import LogItem
from aliyun.log.logclient import LogClient
from aliyun.log.getlogsrequest import GetLogsRequest
from aliyun.log.getlogsrequest import GetProjectLogsRequest
from aliyun.log.putlogsrequest import PutLogsRequest
from aliyun.log.listtopicsrequest import ListTopicsRequest
from aliyun.log.listlogstoresrequest import ListLogstoresRequest
from aliyun.log.gethistogramsrequest import GetHistogramsRequest
from aliyun.log.index_config import *
from aliyun.log.logtail_config_detail import *
from aliyun.log.machine_group_detail import *
from aliyun.log.acl_config import *
ifname=='main':
token = None
endpoint = "http://cn-hangzhou.log.aliyuncs.com"
accessKeyId = '*****'
accessKey = '*****'
client = LogClient(endpoint, accessKeyId, accessKey,token)
logstore = "meta"
In the query statement, specify two Logstores, the query time ranges of both Logstores, and the key to join the Logstores.
req = GetProjectLogsRequest(project,"select count(1) from sls_operation_log s join meta m on sdate >'2018-04-10 00:00:00' and
sdate < '2018-04-11 00:00:00' and mdate >'2018-04-23 00:00:00' and mdate <'2018-04-24 00:00:00' and s.projectid = cast(m.ikey a
s varchar)");
res = client.get_project_logs(req)
res.log_print();
exit(0)

? Note

For more information about the syntax and usage examples of JOIN clauses, see JOIN clause.

4.4.10.33. UNNEST clause

This topic describes the syntax of UNNEST clauses.

Scenario

The value of a column in log data is stored as a primitive data type, such as string or number. In some cases, the value of a column may be of a complex data type, such as array, map, or JSON. When you query and analyze logs that contain fields whose values are of the preceding types, you can use an UNNEST clause to expand the field values into multiple rows for analysis.

Example:

__source_: 1.1.1.1_tag_:_hostname_: vm-req-170103232316569850-tianchi111932.tc_topic_: TestTopic_4array_column: [1,2,3]double_column: 1 .23map_column: {"a":1, "b":2}text_column: Product

The value of the array_column field is an array. To obtain the sum of all elements in the value of the array_column field, you must traverse all elements of each array.

Syntax of UNNEST clauses

Syntax	Description
unnest(array) as table_alias(column_name)	Expands an array into multiple rows. column_name specifies the column name of the rows.
<pre>unnest(map) as table(key_name, value_name)</pre>	$ \begin{array}{llllllllllllllllllllllllllllllllllll$

(2) Note An UNNEST clause is applicable only to arrays or maps. If you want to expand a string, you must convert the string to JSON data. Then, you can use the cast(json_parse(array_column) as array(bigint)) syntax to convert the JSON data to an array or a map.

Traverse the elements of an array

Use an UNNEST clause to expand an array into multiple rows. The rows are stored in a table named t. The column name of the rows is referenced as a.

* | select array_column, a from log, unnest(cast(json_parse(array_column) as array(bigint))) as t(a)

When the elements in an array are traversed, you can also use other SQL syntax to query and analyze data. Examples:

· Calculate the sum of the elements in an array:

* | select sum(a) from log, unnest(cast(json_parse(array_column) as array(bigint))) as t(a)

• Use a GROUP BY clause to group the elements in an array by column name:

* | select a, count(1) from log, unnest(cast(json_parse(array_column) as array(bigint))) as t(a) group by a

Traverse the elements of a map

· Traverse the elements of a map:

* | select map_column, a, b from log, unnest(cast(json_parse(map_column) as map(varchar, bigint))) as t(a,b)

• Use a GROUP BY clause to group the elements in a map by key:

* | select key, sum(value) from log, unnest(cast(json_parse(map_column) as map(varchar, bigint))) as t(key, value) GROUP BY key

Visualize the results of the histogram and numeric_histogram functions

histogram

The histogram function is similar to the count group by syntax. For more information, see Map functions and operators.

The histogram function returns JSON data that cannot be visualized. The following example shows a query statement:

* | select histogram(method)

To visualize the logs that contain the method field, you can use an UNNEST clause to expand the JSON data that is returned by the histogram function into multiple rows. The following example shows a query statement:

* | select key, value from (select histogram(method) as his from log), unnest(his) as t(key, value)

numeric_histogram

The numeric histogram function is used to compute the approximate histogram of a specified field based on the number of histogram columns specified by the bucket parameter. This function is equivalent to the GROUP BY clause that is used to group data by numeric value column. For more information, see Approximate functions.

* | select numeric histogram(10, Latency)

To visualize the result of the numeric_histogram function, execute the following query statement:

* | select key, value from (select numeric_histogram(10, Latency) as his from log), unnest(his) as t(key, value)

4.4.11. Machine learning syntax and functions

4.4.11.1. Overview

Log Service provides the machine learning feature that supports multiple algorithms and calling methods. You can use SELECT statements and machine learning functions to call machine learning algorithms and analyze the characteristics of one or more fields within a specific period of time.

Log Service offers various time series analysis algorithms. You can call these algorithms to solve problems that are related to time series data. For example, you can predict time series, detect time series anomalies, decompose time series, and cluster multiple time series. In addition, the algorithms are compatible with standard SQL functions. This simplifies the usage of the algorithms and improves the efficiency of troubleshooting.

Features

- Supports various smooth operations on single-time series data.
- Supports algorithms that are used for the prediction, anomaly detection, change point detection, inflection point detection, and multi-period estimation of single-time series data.
- Supports decomposition operations on single-time series data.
- · Supports various clustering algorithms for multi-time series data.
- Supports multi-field pattern mining based on the sequence of numeric data or text.

Limits

When you use the machine learning feature of Log Service, take note of the following limits:

- The specified time series data must be sampled based on the same interval.
- The specified time series data cannot contain data that is repeatedly sampled from the same point in time.
- The processing capacity cannot exceed the maximum capacity. The following table describes the limits.

Item	Description
Capacity of the time-series data processing	Data can be collected from a maximum of 150,000 consecutive points in time. If the data volume exceeds the processing capacity, you must aggregate the data or reduce the sampling amount.
Capacity of the density-based clustering algorithm	A maximum of 5,000 time series curves can be clustered at a time. Each curve cannot contain more than 1,440 points in time.

Capacity of the hierarchical clustering algorithm

A maximum of 2,000 time series curves can be clustered at a time. Each curve cannot contain more than 1,440 points in time.

Machine learning functions

Туре	Function	Description
	ts_smooth_simple	Uses the Holt-Winters forecasting algorithm to filter time series data.
Smooth functions	ts_smooth_fir	Uses a finite impulse response (FIR) filter to filter time series data.
	ts_smooth_iir	Uses an infinite impulse response (IIR) filter to filter time series data.
Multi-period estimation functions	ts_period_detect	Estimates time series data by period.
Change point detection functions	ts_cp_detect	Detects the intervals in which data has different statistical features. The interval endpoints are change points.
	ts_breakout_detect	Detects the points in time at which data dramatically changes.
Maximum value detection function	ts_find_peaks	Detects the local maximum value of time series data in a specified window.
	ts_predicate_simple	Uses default parameters to model time series data, predict time series data, and detect anomalies.
	ts_predicate_ar	Uses an autoregressive (AR) model to model time series data, predict time series data, and detect anomalies.
Prediction and anomaly detection	ts_predicate_arma	Uses an autoregressive moving average (ARMA) model to model time series data, predict time series data, and detect anomalies.
functions	ts_predicate_arima	Uses an autoregressive integrated moving average (ARIMA) model to model time series data, predict time series data, and detect anomalies.
	ts_regression_predict	Predicts the trend for a single periodic time series.
	ts_anomaly_filter	Filters the anomalies that are detected from multiple time series curves based on the custom anomaly mode. The anomalies are detected during the anomaly detection. This function helps you find abnormal curves at the earliest opportunity.
Time series decomposition function	ts_decompose	Uses the Seasonal and Trend decomposition using Loess (STL) algorithm to decompose time series data.
	ts_density_cluster	Uses a density-based clustering method to cluster multiple time series.
Time series clustering functions	ts_hierarchical_cluster	Uses a hierarchical clustering method to cluster multiple time series.
	ts_similar_instance	Queries time series curves that are similar to a specified time series curve.
Frequent pattern statistics function	pattern_stat	Mines representative combinations of attributes among the given multi-attribute field samples to obtain the frequent pattern in statistical patterns.
Differential pattern statistics function	pattern_diff	Identifies the pattern that causes differences between two collections in specified conditions.
Root cause analysis function	rca_kpi_search	Analyzes the subdimension attributes that cause the anomalies of a monitoring metric.
Correlation analysis functions	ts_association_analysis	Identifies the metrics that are correlated to a specified metric among multiple observed metrics in the system.
	ts_similar	Identifies the metrics that are correlated to specified time series data among multiple observed metrics in the system.
Kernel density estimation function	kernel_density_estimation	Uses the smooth peak function to fit the observed data points. In this way, the function simulates the real probability distribution curve.

4.4.11.2. Smooth functions

This topic describes the smooth functions that you can use to smooth and filter specified time series curves. Filtering is the first step to discover the shapes of time series curves.

Functions

Function	Description
ts_smooth_simple	Uses the Holt-Winters forecasting algorithm to filter time series data. This function is the default smooth function.
ts_smooth_fir	Uses a finite impulse response (FIR) filter to filter time series data.
ts_smooth_iir	Uses an infinite impulse response (IIR) filter to filter time series data.

ts_smooth_simple

• Syntax

select ts_smooth_simple(x, y)

• The following table describes the parameters in the function.

Parameter Description

User Guide-Log Service

Cloud Defined Storage

x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
у	The sequence of numeric data at a specific point in time.	None

Example

Query statement

* | select ts_smooth_simple(stamp, value) from (select __time__ - __time__ % 120 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)

• Query result



• The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	The UNIX timestamp of the data. Unit: seconds.
Mentical suis	src	The unfiltered data.
	filter	The filtered data.

ts_smooth_fir

• Syntax

- If you cannot determine filter parameters, use the built-in window parameters in the following statement:
 select ts_smooth_fir(x, y,winType,winSize)
- If you can determine filter parameters, you can specify the parameters as needed in the following statement:
 select ts_smooth_fir(x, y,array[])

• The following table describes the parameters in the function.

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
У	The sequence of numeric data at a specific point in time.	None
winType	The type of the window that you want to use to filter data.	Valid values: • rectangle: rectangle window • hanning: hanning window • hamming: hamming window • blackman: blackman window ③ Note We recommend that you set the winType parameter to rectangle for better display effects.
winSize	The length of the filter window.	The value is of the long type. Valid values: 2 to 15.
array[]	The parameter that you want to use for FIR filtering.	The value is an array and the sum of the elements in the array is 1. Example: array[0.2, 0.4, 0.3, 0.1].

• Example 1

Query statement

* | select ts_smooth_fir(stamp, value, 'rectangle', 4) from (select __time__ - __time__ % 120 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)



• Example 2

Query statement

* | select ts_smooth_fir(stamp, value, array[0.2, 0.4, 0.3, 0.1]) from (select __time__ - __time__ % 120 as stamp, avg(v) as value from 1 og GROUP BY stamp order by stamp)

• Query result



• The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	The UNIX timestamp of the data. Unit: seconds.
Vertical avia	src	The unfiltered data.
Vertical axis	filter	The filtered data.

ts_smooth_iir

• Syntax

select ts_smooth_iir(x, y, array[], array[])

• The following table describes the parameters in the function.

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
У	The sequence of numeric data at a specific point in time.	None
array[]	The parameter that you want to use for IIR filtering in terms of x $_{\rm i\cdot}$	The value is an array and the sum of the elements in the array is 1. The length of the array ranges from 2 to 15. Example: array[0.2, 0.4, 0.3, 0.1].
array[]	The parameter that you want to use for IIR filtering in terms of y $_{\rm i-1}$	The value is an array and the sum of the elements in the array is 1. The length of the array ranges from 2 to 15. Example: array[0.2, 0.4, 0.3, 0.1].

• Example

Query statement

* | select ts_smooth_iir(stamp, value, array[0.2, 0.4, 0.3, 0.1], array[0.4, 0.3, 0.3]) from (select __time__ - __time__ % 120 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)



• The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	The UNIX timestamp of the data. Unit: seconds.
Martial ania	src	The unfiltered data.
	filter	The filtered data.

4.4.11.3. Multi-period estimation functions

This topic describes multi-period estimation functions that you can use to estimate the periodicity of time series data distributed in different time intervals. This topic also describes how to extract the periodicity by using a series of operations such as Fourier transform (FT).

Functions

Function	Description
ts_period_detect	Estimates the periodicity of time series data that is distributed in different time intervals.
ts_period_classify	Uses FT to calculate the periodicity of specified time series curves. This function can be used to identify periodic curves.

ts_period_detect

Syntax

select ts_period_detect(x,y,minPeriod,maxPeriod)

The following table describes the parameters in the function.

Parameter	Description
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis. Each point in time is a UNIX timestamp. Unit: seconds.
У	The sequence of numeric data at a specific point in time.
minPeriod	The ratio of the minimum length of the estimated period to the total length of the time series data. The value of this parameter is of the float type. Valid values: (0,1].
maxPeriod	The ratio of the maximum length of the estimated period to the total length of the time series data. The value of this parameter is of the float type. Valid values: (0,1]. (?) Note The value of the <i>maxPeriod</i> parameter must be greater than the value of the <i>minPeriod</i> parameter.

Example

• Query statement

* | select ts_period_detect(stamp, value, 0.2, 1.0) from (select __time__ - __time__ % 120 as stamp, avg(v) as value from log GROUP BY stam p order by stamp)

• Query result



Description

Display item

> Document Version: 20240703

Cloud Defined Storage

period_id	An array with a length of 1. The element in the array indicates the sequence number of the period. The array [0] indicates the original time series curve.
time_series	The sequence of timestamps.
data_series	The sequence of data at each timestamp.If the value of period_id is 0, the function returns the original time series data.If the value of period_id is not 0, the function returns filtered time series data.

ts_period_classify

Syntax

select ts_period_classify(stamp,value,instanceName)

The following table describes the parameters in the function.

Parameter	Description
stamp	The time sequence. Points in time are sorted in ascending order along the horizontal axis. Each point in time is a UNIX timestamp. Unit: seconds.
value	The sequence of numeric data at a specific point in time.
instanceName	The name of the time series curve.

Example

• Query statement

* and h : nu2h05202.nu8 | select ts_period_classify(stamp, value, name) from log

• Query result

line_name	≑⊂, prob	≑⊂, type	\$C
asg-2zejojn6zf5ewg188pg5	1.0	-1.0	>
asg-bp1j8snc92p6v5pptgpj	0.07203669207039314	0.0	
asg-wz99hse7u4ubopo5dt9o	0.0	0.0	
asg-bp18oqni0gq96vy85te4	0.05590892692207093	0.0	

The following table describes the display items.

Display item	Description
line_name	An array with a length of 1. The element in the array indicates the sequence number of the period. The array [0] indicates the original time series curve.
prob	The ratio of the number of values within the primary period to the total number of values on the time series curve. Valid values: $[0, 1]$. You can set the value to 0.15 for testing.
type	 The type of the curve. Valid values: -1: The time series curve has a length of less than 64 points. -2: The time series curve has a failure rate of higher than 20%. 0: The time series curve is periodic.

4.4.11.4. Change point detection functions

This topic describes the change point detection functions that you can use to detect the change points in time series data.

Change point detection functions can detect the following two kinds of change points:

Changes in statistical features within a specific period of time

Anomalies in time series data

Functions

Function	Description
ts_cp_detect	Detects the intervals in which data has different statistical features. The interval endpoints are change points.
ts_breakout_detect	Detects the points in time at which data dramatically changes.

ts_cp_detect

Syntax

select ts_cp_detect(x, y, minSize)

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
У	The sequence of numeric data at a specific point in time.	None

minSize	The minimum length of time series data within a consecutive interval.	The minimum length is 3. The maximum length cannot exceed one tenth of the length of the specified time series data. Default value: 10.
		Vilde. 10.

Example

• Query statement

* | select ts_cp_detect(stamp, value, 3) from (select __time__ - __time__ % 10 as stamp, avg(v) as value from log GROUP BY stamp order by st amp)

• Query result



The following table describes the display items.

	Description
unixtime	The timestamp of the data. Unit: seconds. Example: 1537071480.
src	The unfiltered data. Example: 1956092.7647745228.
prob	The probability that a point in time is a change point. Valid values: 0 to 1.
	unixtime src prob

ts_breakout_detect

Syntax

select ts_breakout_detect(x, y, winSize)

The following table describes the parameters in the function.

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
У	The sequence of numeric data at a specific point in time.	None
winSize	The minimum length of time series data within a consecutive interval.	The minimum length is 3. The maximum length cannot exceed one tenth of the length of the specified time series data. Default value: 10.

Example

Query statement

* | select ts_breakout_detect(stamp, value, 3) from (select __time__ - __time__ % 10 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)

Query result



The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	The timestamp of the data. Unit: seconds. Example: 1537071480.
Menticel cuit	src	The unfiltered data. Example: 1956092.7647745228.
vertical axis	prob	The probability that a point in time is a change point. Valid values: 0 to 1.

4.4.11.5. Maximum value detection function

This topic describes the maximum value detection function that you can use to detect the local maximum value of time series data in a specified window

ts_find_peaks

```
Syntax
```

select ts_find_peaks(x, y, winSize)

The following table describes the parameters in the function.

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
У	The sequence of numeric data at a specific point in time.	None
winSize	The minimum length of the detection window.	The value of this parameter is of the long type and ranges from 1 to the length of time series data. We recommend that you set this parameter to one tenth of the actual data length.

Example

• Query statement

* and h : nu2h05202.nu8 and m: NET | select ts_find_peaks(stamp, value, 30) from (select __time__ - __time__ % 10 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)



The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	The timestamp of the data. Unit: seconds. Example: 1537071480.
	src	The unfiltered data. Example: 1956092.7647745228.
Vertical axis	peak_flag	Indicates whether the numeric value at a point in time is the maximum value.Valid values:1.0: The numeric value at the point in time is the maximum value.0.0: The numeric value at the point in time is not the maximum value.

4.4.11.6. Prediction and anomaly detection functions

Prediction and anomaly detection functions predict the trend of time series curves and identify the Ksigma and quantiles of the errors between a predicted curve and an actual curve. You can use the functions to detect anomalies.

Functions

Function	Description
ts_predicate_simple	Uses default parameters to model time series data, predict time series data, and detect anomalies.
ts_predicate_ar	Uses an autoregressive (AR) model to model time series data, predict time series data, and detect anomalies.
ts_predicate_arma	Uses an autoregressive moving average (ARMA) model to model time series data, predict time series data, and detect anomalies.
ts_predicate_arima	Uses an autoregressive integrated moving average (ARIMA) model to model time series data, predict time series data, and detect anomalies.
ts_regression_predict	Predicts the trend for a single periodic time series. Scenario: You can use this function to predict metering data, network traffic, financial data, and different business data that follows certain rules.
ts_anomaly_filter	Filters the anomalies that are detected from multiple time series curves based on the custom anomaly mode. The anomalies are detected during the anomaly detection. This function helps you find abnormal curves at the earliest opportunity.

ts_predicate_simple

Syntax

select ts_predicate_simple(x, y, nPred, isSmooth)

The following table describes the parameters in the function.

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
У	The sequence of numeric data at a specific point in time.	None
nPred	The number of points for prediction.	The value is of the long type. This value must be equal to or greater than 1.
isSmooth	Specifies whether to filter the raw data.true: The raw data is filtered.false: The raw data is not filtered.	The value is of the Boolean type. Default value: true.

Example

Query statement

* | select ts_predicate_simple(stamp, value, 6) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GROUP BY stamp orde r by stamp)

• Query result



The following table describes the display items.

Display item		Description
Horizontal axis unixtime		The UNIX timestamp of the data. Unit: seconds.
	src	The raw data.
	predict	The predicted data.
Vertical axis	upper	The upper limit of the prediction. The confidence level is 0.85. This value cannot be modified.
	lower	The lower limit of the prediction. The confidence level is 0.85. This value cannot be modified.
	anomaly_prob	The probability that the point is an anomaly. Valid values: 0 to 1.

ts_predicate_ar

select ts_predicate_ar(x, y, p, nPred, isSmooth)

The following table describes the parameters in the function.

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
У	The sequence of numeric data at a specific point in time.	None
p	The order of the AR model.	The value is of the long type. Valid values: 2 to 8.
nPred	The number of points for prediction.	The value is of the long type. Valid values: 1 to 5p.
isSmooth	Specifies whether to filter the raw data.true: The raw data is filtered.false: The raw data is not filtered.	The value is of the Boolean type. Default value: true.

Query statement

* | select ts_predicate_ar(stamp, value, 3, 4) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GROUP BY stamp order b y stamp)

③ Note The result is similar to the result that is returned by the ts_predicate_simple function. For more information, see ts_predicate_simple.

ts_predicate_arma

Syntax

Syntax

select ts_predicate_arma(x, y, p, q, nPred, isSmooth)

The following table describes the parameters in the function.

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
У	The sequence of numeric data at a specific point in time.	None
p	The order of the AR model.	The value is of the long type. Valid values: 2 to 100.
q	The order of the ARMA model.	The value is of the long type. Valid values: 2 to 8.
nPred	The number of points for prediction.	The value is of the long type. Valid values: 1 to 5p.
isSmooth	Specifies whether to filter the raw data. • true: The raw data is filtered. • false: The raw data is not filtered.	The value is of the Boolean type. Default value: true.

Query statement

* | select ts_predicate_arma(stamp, value, 3, 2, 4) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GROUP BY stamp or der by stamp)

🕐 Note The result is similar to the result that is returned by the ts_predicate_simple function. For more information, see ts_predicate_simple.

ts_predicate_arima

Syntax

select ts_predicate_arima(x, y, p, d, q, nPred, isSmooth)

The following table describes the parameters in the function.

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
У	The sequence of numeric data at a specific point in time.	None
p	The order of the AR model.	The value is of the long type. Valid values: 2 to 8.
d	The order of the ARIMA model.	The value is of the long type. Valid values: 1 to 3.
q	The order of the ARMA model.	The value is of the long type. Valid values: 2 to 8.
nPred	The number of points for prediction.	The value is of the long type. Valid values: 1 to 5p.
isSmooth	Specifies whether to filter the raw data.true: The raw data is filtered.false: The raw data is not filtered.	The value is of the Boolean type. Default value: true.

Query statement

* | select ts_predicate_arima(stamp, value, 3, 1, 2, 4) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)

() Note The result is similar to the result that is returned by the ts_predicate_simple function. For more information, see ts_predicate_simple.

ts_regression_predict

Syntax

select ts_regression_predict(x, y, nPred, algotype, processType)

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
у	The sequence of numeric data at a specific point in time.	None
nPred	The number of points for prediction.	The value is of the long type. Valid values: 1 to 500.

algotype	 The type of the algorithm used for prediction. Valid values: origin: uses the Gradient Boosted Regression Tree (GBRT) algorithm for prediction. forest: uses the GBRT algorithm for prediction based on the trend component decomposed by Seasonal and Trend decomposition using Loess (STL), and then uses the additive model to calculate the sum of the decomposed components and obtains the predicted data. linear: uses the Linear Regression algorithm for prediction based on the trend components decomposed by STL, and then uses the additive model to calculate the sum of the decomposed on the trend components decomposed by STL. 	None
processType	Specifies whether to preprocess the data. Valid values:0: No additional data preprocessing is performed.1: Abnormal data is removed before prediction.	None

Example

• Query statement

* and h : nu2h05202.nu8 and m: NET | select ts_regression_predict(stamp, value, 200, 'origin') from (select __time__ - __time__ % 60 as sta mp, avg(v) as value from log GROUP BY stamp order by stamp)

• Query result



The following table describes the display items.

Display item		Description	
Horizontal axis	unixtime	The UNIX timestamp of the data. Unit: seconds.	
Vortical axis	src	The raw data.	
	predict	The predicted data.	

ts_anomaly_filter

Syntax

select ts_anomaly_filter(lineName, ts, ds, preds, probs, nWatch, anomalyType)

The following table describes the parameters in the function.

Parameter	Description	Value
lineName	The name of each curve. The value is of the varchar type.	None
ts	The time sequence of the curve, which indicates the time of the current curve. The value of this parameter is an arra time of the double type. The points in in ascending order.	
ds	The value of this parameter is an arra of the double type. The length of the same as the length of the value of th	
preds	The predicted value sequence of the curve.	The value of this parameter is an array of data points of the double type. The length of the value is the same as the length of the value of the ts parameter.
probs	The sequence of anomaly detection results of the curve.	The value of this parameter is an array of data points of the double type. The length of the value is the same as the length of the value of the ts parameter.
nWatch	The number of the actual values that are recently observed on the curve. The value is of the long type. This value must be less than the number of points in time on the curve.	The value is of the long type.
anomalyType	The type of anomaly that you want to filter. Valid values: • 0: all anomalies • 1: positive anomalies • -1: negative anomalies	The value is of the long type.

Example

- Query statement
 - \star | select res.name, res.ts, res.ds, res.preds, res.probs from (
 - select ts_anomaly_filter(name, ts, ds, preds, probs, cast(5 as bigint), cast(1 as bigint)) as res
 - from (
 - select name, res[1] as ts, res[2] as ds, res[3] as preds, res[4] as uppers, res[5] as lowers, res[6] as probs from (
 - select name, array_transpose(ts_predicate_ar(stamp, value, 10)) as res
 - from (

select name, stamp, value from log where name like '%asg-%') group by name)));

• Query result

name	ts		I	ds	I	preds	I	probs	1
			- L						1
asg-bp1hylzdi2wx7civ0ivk	[1.5513696E9, 1.5513732E	9, 1.5513768E9, 1.5513804E9]	1	[1,2,3,NaN]	I	[1,2,3,4]	I	[0,0,1,NaN]	1

4.4.11.7. Time series decomposition function

This topic describes the time series decomposition function that you can use to decompose time series curves and show the trend and periodicity of curves.

ts_decompose

Syntax

```
select ts_decompose(x, y)
```

The following table describes the parameters in the function.

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
у	The sequence of numeric data at a specific point in time.	None

Example

• Query statement

* | select ts_decompose(stamp, value) from (select __time__ - __time__ % 60 as stamp, avg(v) as value from log GROUP BY stamp order by stamp)

• Query result



Display item		Description	
Horizontal axis unixtime		The UNIX timestamp of the data. Unit: seconds.	
	src	The raw data.	
Vortical axis	trend	The decomposed data that indicates the trend of the time series data.	
Vertical axis	season	The decomposed data that indicates the periodicity of the time series data.	
	residual	The residual data that is decomposed from the time series data.	

4.4.11.8. Time series clustering functions

You can use time series clustering functions to cluster data of multiple time series and obtain different curve shapes. Then, you can use the data to find the cluster center and identify curves with shapes that are different from other curve shapes in the cluster.

Functions

Function	Description
ts_density_cluster	Uses a density-based clustering method to cluster multiple time series.
ts_hierarchical_cluster	Uses a hierarchical clustering method to cluster multiple time series.
ts_similar_instance	Queries time series curves that are similar to a specified time series curve.

ts_density_cluster

Syntax

select ts_density_cluster(x, y, z)

The following table describes the parameters in the function.

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
у	The sequence of numeric data at a specific point in time.	None
Z	The name of the curve that corresponds to the data at a specified point in time.	The value of this parameter is a string. Example: machine01.cpu_usr.

Example

• Query statement

* and (h: "machine_01" OR h: "machine_02" OR h : "machine_03") | select ts_density_cluster(stamp, metric_value,metric_name) from (select _ _time__ - _time__ % 600 as stamp, avg(v) as metric_value, h as metric_name from log GROUP BY stamp, metric_name order BY metric_name, stamp)

Query result



The following table describes the display items.

Display item	Description
cluster_id	The category of the cluster. The value -1 indicates that the cluster is not categorized in a cluster center.
rate	The proportion of instances in the cluster.
time_series	The timestamp sequence of the cluster center.
data_series	The data sequence of the cluster center.
instance_names	The instances that are included in the cluster center.
sim_instance	The name of an instance in the cluster.

ts_hierarchical_cluster

Syntax

. . .

select ts_hierarchical_cluster(x, y, z)

User Guide-Log Service

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
У	The sequence of numeric data at a specific point in time.	None
Z	The name of the curve that corresponds to the data at a specified point in time.	The value of this parameter is a string. Example: machine01.cpu_usr.

Example

• Query statement

* and (h: "machine_01" OR h: "machine_02" OR h : "machine_03") | select ts_hierarchical_cluster(stamp, metric_value, metric_name) from (
select __time__ - __time__ % 600 as stamp, avg(v) as metric_value, h as metric_name from log GROUP BY stamp, metric_name order BY
metric_name, stamp)

• Query result



The following table describes the display items.

Display item	Description
cluster_id	The category of the cluster. The value -1 indicates that the cluster is not categorized in a cluster center.
rate	The proportion of instances in the cluster.
time_series	The timestamp sequence of the cluster center.
data_series	The data sequence of the cluster center.
instance_names	The instances that are included in the cluster center.
sim_instance	The name of an instance in the cluster.

ts_similar_instance

Syntax

select ts_similar_instance(x, y, z, instance_name, topK, metricType)

Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
У	The sequence of numeric data at a specific point in time.	None
Ζ	The name of the curve that corresponds to the data at a specified point in time.	The value of this parameter is a string. Example: machine01.cpu_usr.

instance_name	The name of a specified curve that you want to query.	The value is of this parameter is a string. Example: machine01.cpu_usr. ① Important You must specify an existing curve.
topK	The maximum number of curves that are similar to the specified curve can be returned.	None
metricType	{ <code>'shape', 'manhattan', 'euclidean'}</code> . The metric used to measure the similarity between time series curves.	None

Query statement

* and m: NET and m: Tcp and (h: "nu4e01524.nu8" OR h: "nu2i10267.nu8" OR h: "nu4q10466.nu8") | select ts_similar_instance(stamp, metric_value, metric_name, 'nu4e01524.nu8') from (select __time__ - __time__ % 600 as stamp, sum(v) as metric_value, h as metric_name from 1 og GROUP BY stamp, metric_name order BY metric_name, stamp)

The following table describes the display items.

Display item	Description
instance_name	The list of metrics that are similar to the specified metric.
time_series	The timestamp sequence of the cluster center.
data_series	The data sequence of the cluster center.

4.4.11.9. Frequent pattern statistics function

The frequent pattern statistics function combines representative attributes in a specified multi-attribute field sample.

pattern_stat

Syntax:

select pattern_stat(array[col1, col2, col3], array['col1_name', 'col2_name', 'col3_name'], array[col5, col6], array['col5_name', 'col6_name'],
support_score, sample_ratio)

The following table lists the parameters of the function.

Parameter	Description	Value
array[col1, col2, col3]	A column of character values.	An array of values, for example, array[clientIP, sourceIP, path, logstore].
array['col1_name', 'col2_name', 'col3_name']	The field names of the character values.	An array of field names, for example, array['clientIP', 'sourceIP', 'path', 'logstore'].
array[col5, col6]	A column of numeric values.	An array of values, for example, array[Inflow, OutFlow].
array['col5_name', 'col6_name']	The field names of the numeric values.	An array of field names, for example, array['Inflow', 'OutFlow'].
support_score	The support ratio of samples for pattern mining.	The value is of the DOUBLE data type. Value range: (0,1].
sample_ratio	The sampling ratio. The default value is 0.1, which indicates that only 10% of the total samples are used.	The value is of the DOUBLE data type. Value range: (0,1].

Example:

· Query statement

* | select pattern_stat(array[Category, ClientIP, ProjectName, LogStore, Method, Source, UserAgent], array['Category', 'ClientIP', 'ProjectName', 'LogStore', 'Method', 'Source', 'UserAgent'], array[InFlow, OutFlow], array['InFlow', 'OutFlow'], 0.45, 0.3) limit 1000

• Display item

Display item	Description
count	The number of samples in the current pattern.
support_score	The score of the current pattern. The score indicates the degree to which the current pattern is supported.
pattern	The content of the pattern. The pattern is organized in the format that is defined by the query conditions.

4.4.11.10. Differential pattern statistics function

The differential pattern statistics function analyzes differential patterns of specified multi-field samples based on the specified condition. The function helps you identify the causes of the differences under the current condition at the earliest opportunity.

pattern_diff

Syntax

```
select pattern_diff(array_char_value, array_char_name, array_numeric_value, array_numeric_name, condition,
supportScore,posSampleRatio,negSampleRatio )
```

Parameter	Description	Value
array_char_value	A column of values of the character data type.	The value of this parameter is an array. Example: array[clientlP, sourcelP, path, logstore].
array_char_name	The column names of the values of the character data type.	The value of this parameter is an array. Example: array['clientIP', 'sourceIP', 'path', 'logstore'].
array_numeric_value	A column of numeric values.	The value of this parameter is an array. Example: array[Inflow, OutFlow].
array_numeric_name	The column names of the numeric values.	The value of this parameter is an array. Example: array['Inflow', 'OutFlow'].
condition	The condition that is used to filter data. The value True indicates positive samples and the value False indicates negative samples.	Example: Latency <= 300.
supportScore	The support ratio of positive and negative samples for pattern mining.	The value of this parameter is of the double type. Valid values: (0,1].
posSampleRatio	The sampling ratio of positive samples. Default value: 0.5. This value indicates that 50% of positive samples are collected.	The value of this parameter is of the double type. Valid values: (0,1].
negSampleRatio	The sampling ratio of negative samples. Default value: 0.5. This value indicates that 50% of negative samples are collected.	The value of this parameter is of the double type. Valid values: (0,1].

Example

• Query statement

* | select pattern_diff(array[Category, ClientIP, ProjectName, LogStore, Method, Source, UserAgent], array['Category', 'ClientIP',
'ProjectName', 'LogStore', 'Method', 'Source', 'UserAgent'], array[InFlow, OutFlow], array['InFlow', 'OutFlow'], Latency > 300, 0.2, 0.
1, 1.0) limit 1000

• Display item

Display item	Description
possupport	The support ratio of positive samples for the mined patterns.
posconfidence	The confidence level of the mined patterns in positive samples.
negsupport	The support ratio of negative samples for the mined patterns.
diffpattern	The content of the mined patterns.

4.4.11.11. Root cause analysis function

Log Service provides alerting and analysis capabilities that allow you to analyze and identify anomalies in specific subdimensions of a metric at the earliest opportunity. You can use the root cause analysis function to identify and analyze the subdimension attributes that cause the anomalies.

rca_kpi_search

Syntax

select rca_kpi_search(varchar_array, name_array, real, forecast, level)

The following table describes the parameters in the function.

Parameter	Description	Value
varchar_array	The array of subdimension attributes.	Example: array[col1, col2, col3].
name_array	The array of subdimension attribute names.	Example: array['col1', 'col2', 'col3'].
real	The actual value of each subdimension attribute that is specified by the varchar_array parameter. The value of this parameter is of the double type.	Valid values: all real numbers.
forecast	The predicted value of each subdimension attribute that is specified by the varchar_array parameter. The value of this parameter is of the double type.	Valid values: all real numbers.
level	The number of subdimension attributes identified in the returned root cause set. The value 0 indicates that the function returns all root causes that are found. The value of this parameter is of the double type.	Valid values: [0, number of analyzed subdimensions]. The number of analysis dimensions is based on the length of the array that is specified by the varchar_array parameter.

Example

• Query statement

Use a subquery to obtain the actual value and predicted value of each subdimension attribute, and then call the rca_kpi_search function to analyze the root causes of anomalies.

* not Status:200 |

select rca_kpi_search(

array[ProjectName, LogStore, UserAgent, Method], array['ProjectName', 'LogStore', 'UserAgent', 'Method'], real, forecast, 1)

from (

select ProjectName, LogStore, UserAgent, Method, sum(case when time < 1552436040 then real else 0 end) * 1.0 / sum(case when time < 1552436040</pre>

then 1 else 0 end) as forecast, sum(case when time >=1552436040 then real else 0 end) *1.0 / sum(case when time >= 1552436040 then 1 else 0 end) as real

from (

select __time__ - __time__ % 60 as time, ProjectName, LogStore, UserAgent, Method, COUNT(*) as real from log GROUP by time, ProjectName, LogStore, UserAgent, Method) GROUP BY ProjectName, LogStore, UserAgent, Method limit 100000000)



The following figure shows the structure of the query result.



The following table describes the display items.

Display item	Description
rcSets	The root cause sets. Each value is an array.
rcltems	A root cause set.
kpi	An item in the root cause set. Each item is formatted in an array where each element is of the JSON type. The attr parameter indicates the name of a subdimension. The val parameter indicates the attribute name that corresponds to the subdimension.
nleaf	The number of leaf nodes that a kpi in the root cause set covers in the raw data. ⑦ Note A leaf node is a log entry that contains the finest-grained attributes.
change	The ratio of the number of anomaly changes in the leaf nodes that are covered by a kpi to the total number of anomaly changes in the root cause set at the same point in time.
score	The abnormality score of the current kpi. Valid values: [0,1].

The following example shows the query result that is in the JSON format:

ł			
	"r	cS	Gets": [
	{		
		" 1	cItems": [
		{	
			"kpi": [
			ł
			"attr": "country",
			"val": "*"
			},
			ł
			"attr": "province",
			"val": "*"
			},
			{
			"attr": "provider",
			"val": "*"
			},
			{
			"attr": "domain",
			"val": "download.huya.com"
),
			"attr": "method",
			"val": "*"
			}
			J, 1.1. (1. 110)
			"niedi": 119,
			"change": 0.3160667606279939,
		ı	SCOLE . 0.1445000//09620115
		1	
	3	1	
	1		
}	1		

4.4.11.12. Correlation analysis functions

You can use a correlation analysis function to find the metrics that are correlated with a specified metric or time series data among multiple observed metrics in the system.

Functions

Function	Description
ts_association_analysis	Identifies the metrics that are correlated to a specified metric among multiple observed metrics in the system.
ts_similar	Identifies the metrics that are correlated to specified time series data among multiple observed metrics in the system.

ts_association_analysis

Syntax

select ts_association_analysis(stamp, params, names, indexName, threshold)

The following table describes the parameters in the function.

Parameter	Description	Value
stamp	The UNIX timestamp that is of the long type.	None
params	The metrics that you want to analyze. The value of this parameter is an array. Each element in the array is of the double type.	Example: Latency, QPS, and NetFlow.
names	The names of the metrics that you want to analyze. The value of this parameter is an array. Each element in the array is of the varchar type.	Example: Latency, QPS, and NetFlow.
indexName	The name of the target metric. The value of this parameter is of the varchar type.	Example: Latency.
threshold	The threshold of correlation between the target metric and the metrics that you want to analyze.	Valid values: [0,1].

• Query statement

* | select ts_association_analysis(

time, array[inflow, outflow, latency, status], array['inflow', 'outflow', 'latency', 'status'], 'latency', 0.1) from log;

• Query result

results
['latency', '1.0']
['outflow', '0.6265']
['status', '0.2270']

• Description of the query result

- name: the name of the metric that meets the specified correlation condition of the target metric.
- score: the value of correlation between the returned metric and the target metric. Valid values: [0, 1].

ts_similar

Syntax 1

select ts_similar(stamp, value, ts, ds)
select ts_similar(stamp, value, ts, ds, metricType)

The following table describes the parameters in the function.

Parameter	Description	Value
stamp	The UNIX timestamp that is of the long type.	None
value	The value of the metric that you want to analyze. The value of this parameter is of the double type.	None
ts	The time sequence of the specified time series curve. The value of this parameter is an array. Each element in the array is of the double type.	None
ds	The sequence of numeric data of the specified time series curve. The value of this parameter is an array. Each element in the array is of the double type.	None
metricType	The type of correlation between the measured curves. The value of this parameter is of the varchar type. Valid values: SHAPE, RMSE, PEARSON, SPEARMAN, R2, and KENDALL.	Example: SHAPE .

Syntax 2

select ts_similar(stamp, value, startStamp, endStamp, step, ds)
select ts_similar(stamp, value, startStamp, endStamp, step, ds, metricType)

The following table describes the parameters in the function.

Parameter	Description	Value
stamp	The UNIX timestamp that is of the long type.	None
value	The value of the metric that you want to analyze. The value of this parameter is of the double type.	None
startStamp	The start timestamp of the specified time series curve. The value of this parameter is of the long type.	None
endStamp	The end timestamp of the specified time series curve. The value of this parameter is of the long type.	None
step	The time interval between two adjacent data points in a time series. The value of this parameter is of the long type.	None
ds	The sequence of numeric data of the specified time series curve. The value of this parameter is an array. Each element in the array is of the double type.	None
metricType	The type of correlation between the measured curves. The value of this parameter is of the varchar type. Valid values: SHAPE, RMSE, PEARSON, SPEARMAN, R2, and KENDALL.	Example: SHAPE.

• Query statement

* | select vhost, metric, ts_similar(time, value, 1560911040, 1560911065, 5, array[5.1,4.0,3.3,5.6,4.0,7.2], 'PEARSON') from log group by v host, metric;

• Query result

1	vhost	T	metric	T	score	T
1		Т		Т		I
T	vhost1	T	redolog	T	-0.3519082537204182	T
T	vhost1	T	kv_qps	T	-0.15922168009772697	T
T	vhost1	T	file meta write	T	NaN	T

• Description of the query result

score: the correlation between the analyzed metric and the specified time series curve. Valid values: [-1, 1].

4.4.11.13. Kernel density estimation function

Kernel density estimation (KDE) is a non-parametric way to estimate the probability density function of a random variable. The Kernel density estimation function uses the smooth peak function to fit the observed data points. In this way, the function simulates the real probability distribution curve.

Syntax

select kernel_density_estimation(bigint stamp, double value, varchar kernelType)

Parameters

Parameter	Description
stamp	The Unix timestamp of observed data. Unit: second.
value	The observed value.
kernelType	 box: rectangle window. epanechniov: Epanechnikov curve. gausener: Gaussian curve.

Response

Display item	Description
unixtime	The Unix timestamp of observed data.
real	The observed value.
pdf	The probability of each observed data point.

• Example

Sample statement

- * |
- select date_trunc('second', cast(t1[1] as bigint)) as time, t1[2] as real, t1[3] as pdf from (select kernel_density_estimation(time, num, 'gaussian') as res from (select _time_ - _time_ % 10 as time, COUNT(*) * 1.0 as num from log group by time order by time)), unnest(res) as t(t1) limit 1000 • Response



4.4.12. Scheduled SQL

4.4.12.1. How Scheduled SQL works

Log Service provides the Scheduled SQL feature. You can use the feature to automatically analyze data at regular intervals and aggregate data for storage. You can also use the feature to project and filter data. This topic describes the background information, features, terms, scheduling and running scenarios, and usage notes of Scheduled SQL.

Background information

Time-related data such as logs and metrics can accumulate to excessively large amounts. For example, if 10 million data records are generated per day, a total of approximately 3.6 billion data records are accumulated per year. Long-term data retention requires large storage. If you shorten the data retention period to reduce the required storage, your storage costs can be reduced. However, this may result in the loss of valuable data. In addition, large amounts of data can deteriorate analysis performance.

Data storage and analysis have the following requirements:

- Most metrics are time sensitive. Historical data can have minute or hour precision, but new data must have higher precision.
- Data users such as data operations specialists and data scientists must store full data for analysis.
- The processing of full data and quick response time need to be balanced during data analysis.

To meet the preceding requirements, Log Service provides the Scheduled SQL feature. You can use the feature to compress high-precision historical data to low-precision data and store the compressed data for a long term. After you enable the Scheduled SQL feature, you can change the data retention period of a source Logstore or Metricstore to a smaller value such as 15 days based on your business requirements and change the data retention period of a destination Logstore or Metricstore to permanent. This helps reduce the latency when long-lived data is analyzed and reduce the storage costs.

Features

Scheduled SQL supports SQL-92 syntax and the syntax of Log Service query statements. Scheduled SQL jobs periodically run based on scheduling rules and write the running results to destination Logstores or Metricstores.

- Scheduled data analysis: You can write SQL statements or query statements based on your business requirements to perform scheduled data analysis and store the analysis results to destination Logstores or Metricstores.
- Global aggregation: You can aggregate full and fine-grained data for storage. This process involves lossy compression of data. The storage size and data precision after compression must meet the requirements. Examples:

- If you aggregate 3.6 billion data records for storage based on the second precision, a total of 31.5 million data records are stored, and the storage size is 0.875% of the full data.
- If you aggregate 3.6 billion data records for storage based on the minute precision, a total of 525,000 data records are stored, and the storage size is 0.015% of the full data.
- Projection and filtering: You can filter raw data by field based on specific conditions and store the obtained data to destination Logstores or Metricstores.
- You can also project and filter data by using the data transformation feature, which uses the Domain Specific Language (DSL) syntax. The DSL syntax provides higher extract, transform, and load (ETL) capabilities than the SQL syntax. For more information, see Data transformation basics.

Terms

- Job: Each Scheduled SQL task corresponds to a job. A job includes information such as calculation and scheduling configurations.
- Instance: A Scheduled SQL job generates instances based on scheduling configurations. Each instance performs SQL calculation on raw data and writes the calculation results to the destination Logstore or Metricstore.
- Instance ID: the unique identifier of an instance.
- Creation time: the time when an instance is created. In most cases, an instance is created based on the scheduling rules that you configure. If historical data needs to be processed or if latency exists and needs to be offset, an instance is immediately created.
- Start time: the time when an instance starts to run. If a job is retried, the start time is the time when the last instance of the job starts to run.
- End time: the time when an instance stops running. If a job is retried, the end time is the time when the last instance of the job stops running.Scheduled time: the time for which a job is scheduled. The scheduled time for an instance is generated based on the scheduling rules of the job

regardless of whether the previous instance times out, is delayed, or runs to process historical data. In most cases, the scheduled time for instances that are successively generated is consecutive, and the successive instances can process a complete dataset.

- SQL time window: the time range of data that is analyzed when a Scheduled SQL job runs. Log Service does not analyze data beyond the time range
 when the job runs. An SQL time window is a left-closed and right-open interval that is calculated based on the scheduled time for an instance. An SQL
 time window is independent of the creation time and start time of an instance. For example, if the scheduled time for an instance is 2021/01/01
 10:00:00 and the expression of the SQL time window is [@m 10m, @m), the SQL time window of the instance is [2021/01/01 09:50:00, 2021/01/01
 10:00:00).
- Status: the status of a Scheduled SQL instance. An instance can be in the RUNNING, STARTING, SUCCEEDED, or FAILED state.
- Delayed running: a parameter that you can configure for a Scheduled SQL job. If you set the parameter to N, the instance starts to run after N seconds from the scheduled time. This helps prevent inaccurate calculation results that may be caused by data latency. If you do not need to delay running an instance, you can set the **Delay Task** parameter to 0 Seconds.

For example, if you set the **Specify Scheduling Interval** parameter to **Hourly** and the **Delay Task** parameter to **30** Seconds, 24 instances are generated per day. If the scheduled time for an instance is 2021/4/6 12:00:00, the start time of the instance is 2021/4/6 12:00:30.

Scheduling and running scenarios

Each job can generate multiple instances. Only one instance of a job can be in the RUNNING state at a time, regardless of whether the job is normally scheduled or an instance is retried due to an exception. Multiple instances cannot run at the same time. The following examples illustrate the typical scenarios of scheduling and running:

• Scenario 1: Delay running an instance

The scheduled time for an instance is generated in advance based on the scheduling rules of the job, regardless of whether the instance is delayed from running. If an instance is delayed, the subsequent instances may also be delayed. However, the delay can be gradually offset by running subsequent instances at a higher speed until an instance runs on schedule.

• Scenario 2: Schedule a Scheduled SQL job from a historical point in time

When you create a Scheduled SQL job, you can configure scheduling rules to allow the job to process historical data. When the job is scheduled for the start historical point in time, an instance is generated to process historical data. Then, more instances are generated to process historical data. The instances run in sequence to process historical data until an instance runs on schedule.

• Scenario 3: Schedule a Scheduled SQL job within a specified period of time

If you want to schedule a job to process logs within a period of time, you can specify the period for scheduling. If you specify the end time for scheduling, the job does not generate instances after the last instance runs. The scheduled time for the last instance cannot be the same as or later than the end time for scheduling.

• Scenario 4: Modify scheduling configurations

After you modify the scheduling configurations of a job, the job generates an instance based on the new configurations. If you want to ensure the continuity of SQL time windows among instances, you can modify the SQL time window and scheduling frequency of the scheduling configurations.

Scenario 5: Retry a failed instance

In most cases, a Scheduled SQL job generates instances in chronological order based on scheduled time. If an instance fails to run due to insufficient permissions, nonexistent source Logstore or Metricstore, nonexistent destination Logstore or Metricstore, or invalid SQL syntax, the system allows the instance to automatically retry. If the number of retries exceeds the upper limit that you specify or the instance starts to run.

You can configure alerts for failed instances and manually retry the instances. You can view and retry the instances that are generated within the last seven days. After the instances run, the system changes the status of the instances to SUCCEEDED or FAILED based on the retry results. For more information about how to retry instances, see the "Retry Scheduled SQL instances" section in the "Manage Scheduled SQL jobs" topic.

Usage notes

When you use the Scheduled SQL feature, we recommend that you balance the timeliness and accuracy of data based on your business requirements.

- When data is uploaded to Log Service, latency may exist. In this case, the data for an SQL time window may not be completely uploaded to Log
 Service when an instance is running. To prevent this issue, we recommend that you configure the Delay Task and SQL Time Window parameters
 based on the data collection latency and the maximum result viewing latency allowed for your business. In addition, we recommended that you
 specify values that are slightly earlier than theoretical values to ensure that instances can run as expected.
- To ensure the accuracy of processing results in scenarios in which several unordered data is uploaded, we recommend that you specify minute- or hour-level SQL time windows for jobs.

4.4.12.2. Limits

This topic describes the limits of the Scheduled SQL feature.

Query and analysis

() Important

Scheduled SQL supports only Dedicated SQL.

Item	Description
Number of concurrent analytic statements	Each project supports up to 150 concurrent analytic statements. For example, 150 users can concurrently execute analytic statements in all Logstores of a project.
Data volume	An analytic statement can scan up to 200 billion rows of data.
Applicable scope	You can analyze only the data that is written to Log Service after the log analysis feature is enabled.
Returned result	 By default, an analytic statement returns up to 100 rows of data. Excess data is not returned. If you want an analytic statement to return more data, you can use the LIMIT clause in the statement. Up to 1 million rows of data can be returned. For more information, see LIMIT syntax. Data beyond the range specified by the LIMIT clause is not returned. The volume of data that an analytic statement can return is limited to 20 GB. Excess data is not returned.
Size of a field value	By default, the size of a field value is 2,048 bytes, equivalent to 2 KB. The maximum size of a field value is 16,384 bytes, equivalent to 16 KB. If the size of a field value exceeds 16 KB, the excess data is not involved in analysis. You can modify the maximum size for a field value when you configure indexes. Valid values: 64 to 16384. Unit: bytes. For more information, see Configure indexes.
Timeout period	The maximum timeout period for an analytic statement is 10 minutes.
Number of bits in the mantissa part of a double-type field value	A double-type field value can contain up to 52 bits in the mantissa part. If the mantissa part of a double-type field value contains more than 52 bits, the precision of the field value is compromised.
Fuzzy search	In a fuzzy search, Log Service matches up to 100 words that meet the specified conditions and returns the logs that contain one or more of these words and meet the query conditions.
Inaccurate query result	If query results are inaccurate, no errors are reported. However, the issue is recorded in the instance status information and included in job running records. The recording feature must be manually enabled.
Data latency	If data latency occurs, some data may be missed in query. If the data of a point in time arrives later after the instance for that point in time runs, the data is not included when the next instance runs. For more information, see How do I ensure data accuracy when I execute SQL statements to analyze data?.
Time window	The time window for a single query ranges from 1 minute to 24 hours.
Metastore association	Not supported.

Data write

Item	Description
Write threshold of a Logstore	If the write threshold is exceeded when you write data, the Scheduled SQL job is retried for more than 10 minutes. After the retry time, an error message is returned.

Job running

Item	Description
Timeout period	The maximum timeout period of a job is 1,800 seconds. If the timeout period of a job is exceeded, the job is considered failed. We recommend that you create an alert monitoring task to detect errors and retry failed instances in a timely manner. For more information, see Retry an instance of the Scheduled SQL job
Number of retries	The maximum number of retries for a job is 100. If a job is retried for more than 100 times, the job is considered failed.
Delayed running	You can delay running an instance for up to 120 seconds. For more information about delayed running scenarios, see Scheduling and running scenarios

	The historical running records of a single job can be stored for up to 14 days.
Historical running record	We recommend that you create an alert monitoring task to detect errors and retry failed instances in a timely manner. For more information, see Retry an instance of the Scheduled SQL job

4.4.12.3. Create a Scheduled SQL job

4.4.12.3.1. Process and store data from a Logstore to a Logstore

Log Service provides the Scheduled SQL feature. You can use the feature to analyze data at a scheduled time and aggregate data for storage. You can also use the feature to project and filter data. The Scheduled SQL feature can process data in a source Logstore and store the processed data to a destination Logstore.

Prerequisites

- Data is collected to a source Logstore. For more information, see Data collection overview.
- A destination Logstore is created. For more information, see Create a Logstore.
- Indexes are configured for the source and destination Logstores. For more information, see Configure indexes.

Procedure

- 1. Log on to the Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. On the Log Storage > Logstores tab, click the Logstore that you want to manage.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.

A query statement consists of a search statement and an analytic statement in the **Search statement|Analytic statement** format. For more information, see Log search overview and Log analysis overview.

? Note

This step allows you to preview data before you create a Scheduled SQL job. You can check whether the query statement that you entered is valid and whether the query results contain data.

5. On the Graph tab, click Schedule to Save Analysis Results.

~	1 * se	lect COUN	T(*) as pv						(15 Minutes(Relative)	ve) 🔻 Search & Anal	yze C 🔹 🏠
240												
U	18:03:19		18:04:45	18:06:15	18:07:45	18:09:15	18:10:45	18:12:15	18:13:45	18:15:15	18:16:45	18:18:04
		Log	Entries:4,248 Search S	tatus:The results are accurate. S	icanned Rows:4,248 Search	h Time:111ms Query Results:	(By default, a maximum of	100 rows of data is returned f	or each query. Use the LIM	IIT clause if you want to obtain	more query results.)	
Rav	Logs	Graph	LogReduce									
Ch	art Previe	w				Schedule to Save A	nalysis Results Add to I	New Dashboard Downloa	Hide Settings	Common Setting	Fields Inter	action Events
pv									\$ C	A Chart Types		
424	8									📰 🗞 😭		

6. Create a Scheduled SQL job.

i. In the **Compute Settings** step, configure the following parameters and click **Next**.

Parameter	Description
Job Name	The name of the Scheduled SQL job.
Task Description	The description of the Scheduled SQL job.
Resource Pool	The resource pool that is used for data analysis. Log Service provides an enhanced type of resource pool. The enhanced type of resource pool reuses the computing capability of Dedicated SQL. The enhanced type of resource pool can meet concurrent analysis requirements and isolate resources between Scheduled SQL and your SQL analysis operations in the console. You are charged for the enhanced type of resource pool based on the CPU time that is consumed by your SQL analysis operations. For more information, see Enable Dedicated SQL.
Write Mode	Select Import Data from Logstore to Logstore . The Scheduled SQL feature processes data in the source Logstore and writes the processed data to the destination Logstore.
SQL Code	The query statement. By default, the system displays the statement that you entered in Step4. The preview operation provided for this parameter has the same effect as the preview operation in Step 4. You can click Preview to check whether the query statement is valid and whether the query results contain data. When the Scheduled SQL job runs, Log Service executes this query statement to analyze data.
Target Region	The region where the destination project resides.
Target Project	The name of the destination project, which stores the results of the query statement.
Target Logstore	The name of the destination Logstore, which stores the results of the query statement.
Write Authorization	The method that is used to authorize the Scheduled SQL job to write data to the destination Logstore. Valid values: Custom Role : The Scheduled SQL job assumes a custom role to write the analysis results to the destination Logstore. You must grant the custom role the permissions to write data to the destination Logstore. Then, enter the Alibaba Cloud Resource Name (ARN) of the custom role in the Role ARN field. For more information about how to obtain the ARN, see Create a RAM role and grant the required permissions to the RAM role
SQL Execution Authorization	The method that is used to authorize the Scheduled SQL job to read data from the source Logstore and analyze the data by using query statements in the current project. Valid values: Custom Role : The Scheduled SQL job assumes a custom role to perform the required operations. You must grant the custom role the required permissions. Then, enter the ARN of the custom role in the Role ARN field. For more information, see Create a RAM role and grant the required permissions to the RAM role.

ii. In the Scheduling Settings step, configure the following parameters and click OK.

Parameter	Description
Specify Scheduling Interval	 The frequency at which the Scheduled SQL job is scheduled. An instance is generated each time the Scheduled SQL job is scheduled. This parameter determines the scheduled time for each instance. Valid values: Hourly: The Scheduled SQL job is scheduled every hour. Daily: The Scheduled SQL job is scheduled at a fixed point in time every day. Weekly: The Scheduled SQL job is scheduled at a fixed point in time on a fixed day of each week. Fixed Interval: The Scheduled SQL job is scheduled at a fixed point in time on a fixed day of each week. Fixed Interval: The Scheduled SQL job is scheduled at a fixed point in time on a fixed day of each week. Cron: The Scheduled SQL job is scheduled at an interval that is specified by a cron expression. A cron expression can specify an interval that is accurate to the minute. The cron expression is based on the 24-hour clock. For example, 0 0/1 * * * indicates that the Scheduled SQL job is scheduled at an interval of 1 hour from 00:00. If you need to specify the time zone, select Cron. For more information about common time zones, see Time zones.
Scheduling Time Range	The time range during which the Scheduled SQL job is scheduled. Valid values: • Start at a specified time.: If you select this option, you must specify the point in time at which the first instance of the Scheduled SQL job starts to run. • Within Specific Period: If you select this option, you must specify the time range within which the instances of the Scheduled SQL job can run. () Important If you specify the time range, the instances of the Scheduled SQL job can run only within the time range. After the end time, the Scheduled SQL job no longer generates instances.
SQL Time Window	The time window of logs that are analyzed when the Scheduled SQL job runs. This parameter must be configured together with the Scheduling Time Range parameter. The duration specified by this parameter can be up to five times the duration specified by Specify Scheduling Interval . The start time and end time of the SQL time window must be within 24 hours. For more information, see Syntax of time expressions . For example, Specify Scheduling Interval is set to Fixed Interval 10 Minutes , Start Time is set to 2021-04-01 00:00:00.0 Delay Task is set to 30 Seconds , and SQL Time Window is set to [@m-10m,@m) . In this example, the first instance of the Scheduled SQL job is generated at 00:00:30 to analyze the logs that fall in the time range [23:50:00,00:00:00). For more information, see Scheduling and running scenarios .
SQL Timeout	The threshold of automatic retries if the SQL analysis operation fails. If an instance is retried for a period that exceeds the maximum time that you specify or the number of retries for an instance exceeds the upper limit that you specify, the instance stops retrying and enters the FAILED state. You can manually retry the instance based on the failure cause. For more information, see Retry an instance of the Scheduled SQL job
Delay Task	The number of seconds for which the instance is delayed from the scheduled time. Valid values: 0 to 120. Unit: seconds. If latency exists when data is written to the destination Logstore, you can use this parameter to ensure data integrity.

4.4.12.4. Manage a Scheduled SQL job

On the Scheduled SQL page in the Log Service console, you can view the basic information about a Scheduled SQL job and the instances of the Scheduled SQL job. You can also retry, modify, or delete the Scheduled SQL job.

Prerequisites

A Scheduled SQL job is created. For more information, see Process and store data from a Logstore to a Logstore.

Procedure

- 1. Log on to the Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. In the left-side navigation pane, choose **Jobs > Scheduled SQL**.
- 4. Click the Scheduled SQL job.

View the basic information about the Scheduled SQL job

In the **Basic Information** section, you can view the basic information about the Scheduled SQL job, including Created At, Last Modified At, and Task ID.

Basic Information					
Created At	Sep 5, 2023, 10:13:16		Last Modified At	Sep 5, 2023, 10:13:16	
Source Project/Logstore	oss-log-1	store	Destination Project/Logstore	oss-lo	-test-temp
Job ID 2023	9d62aa9ee43c4de547bef501bce4e1bf		Job Name	sql-1693879996-998386	
Job Description					

View the instances of the Scheduled SQL job

When the Scheduled SQL job is run, Log Service generates instances based on the scheduling interval that you specify. You can view all instances of the Scheduled SQL job in the **Instances** section.

Parameter	Description
Instance ID	The unique identifier of the Scheduled SQL instance.
Job Execution Time	The time range during which the Scheduled SQL instance is run.
SQL Query Range	The time range that you specify for SQL analysis. The instance analyzes the data that is generated within the time range.
Processed Data Size	 The size of the data involved in SQL analysis. The following information is displayed: Processed Rows: the number of logs that the instance reads within the specified SQL time window. This parameter indicates the amount of data that is used in computation. Result Rows: the number of logs in the analysis results. This parameter indicates the amount of data that is written to the destination Logstore or Metricstore. Processed Date Size: the number of bytes of logs that the instance reads within the specified SQL time window. This parameter indicates the amount of data that is used in computation. Rows Written to Destination Logstore: the number of logs that are obtained from the analysis results and written to the destination Logstore or Metricstore. This parameter indicates the amount of data that is written to the destination Logstore or Metricstore.
Status	The status of the Scheduled SQL instance. Valid values: Running, Retrying, Success, and Failed.

Retry an instance of the Scheduled SQL job

If an instance of the Scheduled SQL job is in the Success or Failed state, you can run the instance again. For example, if an instance is in the Failed state due to invalid authorization, you can modify the authorization settings of the Scheduled SQL job and click **Retry**.

() Important

If one or more instances are run to process historical data, you can retry successful instances. However, we recommend that you do not retry successful instances. In most cases, you need to retry only the failed instances of a Scheduled SQL job.

Delete the Scheduled SQL job

If you no longer need to use the Scheduled SQL job, you can click Delete Job in the upper-right corner of the Scheduled SQL page.

\land Warning

After you delete the Scheduled SQL job, the job cannot be restored. Proceed with caution.

Modify the Scheduled SQL job

To modify the settings of the Scheduled SQL job, click **Modify Settings** in the upper-right corner of the Scheduled SQL page. For more information about the parameters, see Process and store data from a Logstore to a Logstore.

4.4.12.5. Query the result data of a Scheduled SQL job

This topic describes how to query the result data of a Scheduled SQL job in a Logstore or a Metricstore.

Prerequisites

A Scheduled SQL job is created. You can create a Scheduled SQL job to process data from a source Logstore and store the processed data to a destination Logstore. For more information, see Process and store data from a Logstore to a Logstore.

Query the result data of the Scheduled SQL job in a Logstore

After you store the result data of the Scheduled SQL job to a Logstore, you can query the result data in the Logstore by using the name of the Scheduled SQL job.

1. Log on to the Log Service console

- 2. Obtain the name of the Scheduled SQL job.
- i. In the Projects section, click the project that you want to manage.
- ii. In the left-side navigation pane, choose Jobs > Scheduled SQL. In the Scheduled SQL list, click the job whose name you want to obtain.
- iii. In the **Basic Information** section, obtain the name of the Scheduled SQL job.

Basic Information					
Created At	Sep 5, 2023, 10:13:16		Last Modified At	Sep 5, 2023, 10:13:16	
Source Project/Logstore	oss-log-1	store	Destination Project/Logstore	oss-lo	-test-temp
Job ID 2023	9d62aa9ee43c4de547bef501bce4e1bf		Job Name	sql-1693879996-998386	
Job Description					

3. Query the result data of the Scheduled SQL job.

- i. Find the Logstore that is used to store the result data of the Scheduled SQL job. For more information, see Query and analyze logs.

The job-name field specifies the name of the Scheduled SQL job. Replace job-name with the name that you obtained in Step2 .
iii. Click 15 Minutes(Relative) and specify a time range for the query.

? Note

If no data is returned, you can specify a longer time range for the query.

iv. Click Search & Analyze.

Query the result data of the Scheduled SQL job in a Metricstore

After you store the result data of the Scheduled SQL job to a Metricstore, you can query the result data in the Metricstore by using the metrics that you specify.

- 1. Log on to the Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. On the **Time Series Storage > Metricstore** tab, click the Metricstore that is used to store the result data.
- 4. Click 15 Minutes(Relative) and specify a time range for the query.

```
⑦ Note
If no data is returned, you can specify a longer time range for the query.
```

5. On the Query Statements tab, select a metric and click Preview.

Select the metric that you specify when you create the Scheduled SQL job.

4.4.12.6. Create a RAM role and grant the required permissions to the RAM role

When a Scheduled SQL job runs, the job executes an SQL statement to analyze data in the source Logstore and stores the query and analysis results in the destination Logstore. You can assign a Resource Access Management (RAM) role to a Scheduled SQL job to complete the preceding operations. This topic describes how to create a RAM role and grant the required permissions to the RAM role.

Procedure

- 1. Log on to the Apsara Uni-manager Management Console as an administrator.
- 2. Create a RAM role and grant the required permissions to the RAM role.
- For more information, see Apsara Uni-manager > User Guide > Enterprise > Permissions > Role Permissions .

When you create a RAM role, select **log.aliyuncs.com** for **Trust Cloud Service**. For more information about the custom policies that grant permissions to a RAM user, see Use custom policies to grant permissions to a RAM user. The following policy grants a RAM role the permissions to read data from the source Logstore and write data to the destination Logstore:



- 3. Obtain the Alibaba Cloud Resource Name (ARN) of the RAM role.
- You can view the **ARN** in the ARN column of the RAM role. Example: acs:ram::14369****3572:role/sls-test . We recommend that you record the ARN. If you use a custom RAM role to create a Scheduled SQL job, you must enter the ARN of the RAM role.

4.4.12.7. Syntax of time expressions

When you create a Scheduled SQL job, you can specify an SQL time window. When a Scheduled SQL job runs, Log Service analyzes only the logs that are generated within the specified SQL time window. This topic describes the syntax of time expressions that are used to specify an SQL time window.

Operators

The following table describes the operators that are supported in time expressions.

Operator	Description
+	The plus sign.
	The minus sign.
@	The operator that is used to round the time down to the nearest time unit. For example, if the time unit is hours, 01:40 is rounded down to the nearest hour 01:00.

A time expression can be in the ±{num}{unit} or @{unit} format. The {num} variable is a positive integer, and the {unit} variable is a time unit.

If a plus sign (+) or a minus sign (-) operator is used, the time expression is in the ±{num}{unit} format. You are not required to specify a value for the {num} variable. If you do not specify a value for the {num} variable, 1 is used by default. For example, if the time expression is -h, 1 hour is subtracted from a time value.

• If the at sign (@) operator is used, the time expression is in the @{unit} format.

Time units

The following table describes the time units that are supported in time expressions.

Time unit	Description
h	Hour
m	Minute
S	Second

Examples

The following table provides time expression examples.

Time expression	Description
-15m@m	Subtracts 15 minutes from a time value and rounds the new time value down to the nearest minute. For example, when a Scheduled SQL job is created, Specify Scheduling Interval is set to Daily00:00 , Delay Task to 30 , and SQL Time Window to [-15m@m,-5m@m) . In this example, the Scheduled SQL job is run at 00:00:30 to analyze the data that is generated within the following time range: [23:45,23:55].
-h@h	Subtracts 1 hour from a time value and rounds the new time value down to the nearest hour. For example, when a Scheduled SQL job is created, Specify Scheduling Interval is set to Daily00:00, Delay Task to 30 , and SQL Time Window to [-h@h,-5m@m) . In this example, the Scheduled SQL job is run at 00:00:30 to analyze the data that is generated within the following time range: [23:00,23:55].
-50m@h	Subtracts 50 minutes from a time value and rounds the new time value down to the nearest hour. For example, when a Scheduled SQL job is created, Specify Scheduling Interval is set to Daily00:00 , Delay Task to 30 , and SQL Time Window to [-50m@h,-5m@m). In this example, the Scheduled SQL job is run at 00:00:30 to analyze the data that is generated within the following time range: [23:00,23:55].
-12h+5m	Subtracts 12 hours from a time value and then adds 5 minutes. For example, when a Scheduled SQL job is created, Specify Scheduling Interval is set to Daily00:00, Delay Task to 30 , and SQL Time Window to [-12h+5m,-5m) . In this example, the Scheduled SQL job is run at 00:00:30 to analyze the data that is generated within the following time range: [12:05,23:55).

4.4.12.8. Time zones

This topic describes the time zone format and common time zones of the Scheduled SQL feature.

Time zone format

The time zones of the Scheduled SQL feature are in the {±Offset hours}{Minutes} format.

- Valid values of offset hours: [-12,+14]
- Valid values of minutes: 00, 30, and 45
- Valid values of time zones: [-1200,+1400]

? Note

Hours and minutes must be formatted as two-digit values.

When you create a Scheduled SQL job, you can select Cron from the drop-down list of the Specify Scheduling Interval field and select a time zone.

Co	mpute Settings			Scheduling Settings
* Specify Scheduling Interval	Cron	~	Enter a cron expression.	0/64
	The minimum pred Hours field is 0 to	cision i 24. Exa	n a cron expression is a minute amples:	. The value range for the
	 0/5 * * * * 0 0/1 * * * 0 18 * * * 0 0 1 * * 	Run Star Run Run	at every 5 minutes t at 00:00 and run at every hou at 18:00 every day at 00:00 on the first day of eve	r ry month
Time Zone	+0800		\vee	

Common time zones

Time zone	Full name
	China Standard Time and Hong Kong Time
	Australian Western Time
	Korea Standard Time
	Malaysia Time
+0800	Philippile Time
	Singapore Time
	Central Indonesian Time
	Ulaanbaatar Time
	Choibalsan Time
10000	Japan Standard Time
10900	Ulaanbaatar Summer Time
+0930	Australian Central Standard Time
+1200	New Zealand Standard Time
+0530	India Standard Time
0000	Greenwich Mean Time
-0000	Western Europen Time
-0400	Atlantic Standard Time
-0500	Eastern Standard Time
-0600	Central Standard Time
-0700	Mountain Standard Time
-0800	Pacific Standard Time

4.4.12.9. FAQ

This topic provides answers to some frequently asked questions about the Scheduled SQL feature.

How do I ensure data accuracy when I execute SQL statements to analyze data?

Data analysis results may be inaccurate due to the following reasons: Latency exists when data is written to Log Service, or the scheduling configurations of instances are invalid.

- A write latency exists when data is written to Log Service. In the scenario with a 5-minute write latency, if an instance whose SQL time window is [12:02:00,12:03:00) runs at 12:03:00, Log Service cannot obtain data for that time window.
- A query latency exists after data is written to Log Service. The latency is generally less than 3 seconds. Some data cannot be obtained even if the latency is low. For example, if an instance whose SQL time window is (12:02:30,12:03:30) runs at 12:03:30, Log Service may fail to obtain the logs that are written to Log Service at 12:03:29 for that time window.
- If the logs that are generated at different points in time are written to Log Service at the same minute, all the logs have the same index based on the storage time. Later logs can have an earlier index. For example, an instance whose SQL time window is [12:02:30,12:03:30] runs at 12:03:30. If two logs are generated at 12:02:20 and 12:02:50 but written to Log Service at 12:02:50, the logs are both indexable by using 12:02:20. In this case, Log Service cannot obtain the logs for the time window [12:02:30,12:03:30].

When you use the Scheduled SQL feature, we recommend that you balance the timeliness and accuracy of data based on your business requirements.

- When data is uploaded to Log Service, latency may exist. In this case, the data for an SQL time window may not be completely uploaded to Log
 Service when an instance is running. To prevent this issue, we recommend that you configure the **Delay Task** and **SQL Time Window** parameters
 based on the data collection latency and the maximum result viewing latency allowed for your business. In addition, we recommended that you
 specify values that are slightly earlier than theoretical values to ensure that instances can run as expected.
- To ensure the accuracy of processing results in scenarios in which several unordered data is uploaded, we recommend that you specify minute- or hour-level SQL time windows for jobs.

How do I prevent failures when I execute SQL statements to analyze data?

- Enter valid SQL statements.
- Configure valid indexes for the fields that you want to analyze. For example, if the query statement is * | select uid you must turn on Enable Analytics for the **uid** field. For more information, see Configure indexes.
- Make sure that your account is granted the required permissions. For example, your account is granted the permissions to execute SQL statements to analyze data and the permissions to read data from Logstores.
- To prevent calculation timeout errors, we recommend that you do not use complex SQL statements or configure a long SQL time window. The timeout period of Log Service is 10 minutes.

Does Log Service check indexes when the calculation results of a Scheduled SQL job are written to the destination Logstore?

Log Service does not check indexes when the calculation results of a Scheduled SQL job are written to the destination Logstore. If you have not configured indexes for the destination Logstore, you cannot execute SQL statements to analyze data. Log consumption and query are not affected.

Before you create a Scheduled SQL job, we recommend that you configure indexes for the destination Logstore.

Does the timeout of an instance affect subsequent execution?

No, the timeout of an instance does not affect subsequent execution. The scheduled time for subsequent instances is in sequence with the scheduled time of the instance that timed out, but the subsequent instances are delayed from creating and running. The delay can be gradually offset by running subsequent instances at a higher speed until an instance runs as scheduled.

In delay offsetting scenarios, if the volume of data that needs to be processed is fixed, a larger scheduling interval indicates a higher offset speed. Examples:

- If 24-hour data needs to be processed and the scheduling interval is 1 minute, a total of 1,440 instances are generated, and each instance runs for 20 seconds.
- If 24-hour data needs to be processed and the scheduling interval is 1 hour, a total of 24 instances are generated, and each instance runs for 2 minutes.

Log Service supports distributed query and analysis and allocates more computing resources if more data needs to be processed.

How do I trace the source of data that is written to a destination Logstore or Metricstore?

By default, the following <u>tag</u> fields are added when the data of a Scheduled SQL job is written to the destination Logstore or Metricstore. You can use these fields to trace the source of the data.

- _tag_:_instance_id_:2b06486746f0cb38-5bffe67493825-1e2606a: the ID of the instance.
- __tag_:__job__:from_now: the name of the job
- __tag_:__project__:ali-sls-etl-staging: the project to which the job belongs.
- _tag_:_schedule_time_:1618474200: the point in time at which the job is scheduled. The time is a UNIX timestamp. Unit: seconds.
- __tag_:__trigger_time_:1618474259: the point in time at which the job is run. The time is a UNIX timestamp. Unit: seconds.

4.4.13. Advanced analysis

4.4.13.1. Optimize queries

This topic describes how to optimize queries to improve query efficiency.

Increase the number of shards

More shards indicate more computing resources and faster computing speed. You can increase the number of shards to ensure that the average number of log entries that are scanned in each shard does not exceed 50 million. You can increase the number of shards by splitting shards. For more information, see Split a shard.

Reduce the query time range and data volume

- A larger time range means a slower query. If you query data within a year or a month, data is computed on a daily basis. To improve the computing speed, you can reduce the query time range.
- Larger data volumes slow down queries. Reduce the amount of data that you want to query as much as possible.

Repeat queries multiple times

If you find that the result of a query is inaccurate, you can repeat the query multiple times. The underlying acceleration mechanism ensures that each query uses the previous query result to analyze data. This way, multiple queries can improve the accuracy of the query result.

Optimize SQL statements for queries

- A time-consuming query statement has the following characteristics:
- Uses GROUP BY clauses to group string-formatted columns.
- Use GROUP BY clauses to group more than five columns.
- Includes operations that generate strings.
- We recommend that you use the following methods to optimize SQL statements for queries:
- · Avoid operations that generate strings if possible.
 - $\circ~$ If you use the date_format function to generate a formatted timestamp, the query is inefficient.
 - * | select date_format(from_unixtime(__time__) , '%H_%i') as t, count(1) group by t
 - If you use the substr() function, strings are generated. We recommend that you use the date_trunc or time_series function in a query statement.
- Avoid using GROUP BY clauses to group string-formatted columns if possible.
 If you use a GROUP BY clause to group strings, a large number of hash calculations are required. The number of the hash calculations account for more than 50% of the total number of calculations. The following example shows two query statements:

* | select count(1) as pv , date_trunc('hour',__time_) as time group by time * | select count(1) as pv , from_unixtime(__time___time_%3600) as time group by __time__-_time_%3600

Query 1 is less efficient than query 2 because query 1 needs to hash strings.

- Query 1 and guery 2 calculate the total number of log entries per hour.
- Query 1 converts the time to a string, for example, 2017-12-12 00:00:00, and then uses a GROUP BY clause to group the string.
- Query 2 calculates the on-the-hour time value, uses a GROUP BY clause to group the result, and then converts the value to a string.
- List fields alphabetically based on the initial letter when you use a GROUP BY clause to group multiple columns.
- For example, you need to query 100 million users who are from 13 provinces.

```
Fast: * | select province,uid,count(1)groupby province,uid
Slow: * | select province,uid,count(1) group by uid,province
```

• Use estimating functions.

Estimating functions provide better performance than accurate calculation. In estimation, accuracy is compromised to an acceptable level to achieve fast calculation.

Fast: * |select approx_distinct(ip)
Slow: * | select count(distinct(ip))

• Specify only the required columns in an SQL statement if possible.

When you use an SQL statement to query data, specify only the required columns to speed up the calculation.

Fast: * | select a,b c
Slow: * | select *

• Specify columns that do not need to be grouped in an aggregate function if possible.

For example, a user ID is associated with a username. Therefore, you can use a GROUP BY clause to group data by userid.

Fast: * | select userid, arbitrary(username), count(1)group by userid Slow: * | select userid, username, count(1) group by userid,username

Avoid using the IN operator if possible.
 Use the OR operator in SQL statements instead of the IN operator if possible.

Fast: key : a or key :b or key:c | select count(1)
Slow: * | select count(1) where key in ('a','b')

4.4.13.2. Use cases

This topic describes some use cases of data analysis in Log Service.

Cases

- Trigger an alert if the error rate exceeds 40% over the last 5 minutes
- Calculate the amount of transferred data and configure alerts
- · Calculate the average latency of traffic data in different sizes
- Obtain the percentages of different results
- · Calculate the number of log entries that meet the query condition

Trigger an alert if the error rate exceeds 40% over the last 5 minutes

Calculate the percentage of 500 Internal Server Error every minute. If the error rate exceeds 40% over the last 5 minutes, an alert is triggered.

Cloud Defined Storage

```
status :500 |
select
  topic ,
 max_by(error_count, window_time) / 1.0 / sum(error_count) as error_ratio,
  sum(error_count) as total_error
FROM (
   select
     __topic__,
count(*) as error_count,
       __time___time__ % 300 as window_time
DM log
   FROM
   group by
       topic ,
     window_time
 )
group by
   _topic
having
 max_by(error_count, window_time) / 1.0 / sum(error_count) > 0.4
  and sum(error_count) > 500
order by
 total_error desc
limit
 100
```

- You can use the following clause to calculate the error rate: max_by(error_count,window_time)/1.0/sum(error_count) as error_ratio .
- You can use the following clause to calculate the total number of 500 Internal Server Error: sum(error_count) as total_error .
- You can use the following clause to query the number of errors every 5 minutes: select _topic_, count(*) as error_count , _time_ _time_ % 300 as window_time from log group by _topic_, window_time .

Calculate the amount of transferred data and configure alerts

Calculate the amount of transferred data every minute. If the amount of transferred data sharply decreases, an alert is triggered. Transferred data counted in the last minute does not cover a full minute. The (max(time) - min(time)) clause is used for normalization to count the average traffic per minute.

```
* |
SELECT
SUM(inflow) / (max(__time__)-min(__time__)) as inflow_per_minute,
date_trunc('minute', __time__) as minute
group by
minute
```

Calculate the average latency of traffic data in different sizes

Distribute traffic data to multiple buckets based on the data size and calculate the average latency of the data in each bucket.

```
* |
select
avg(latency) as latency,
case
when originSize < 5000 then 's1'
when originSize < 20000 then 's2'
when originSize < 500000 then 's3'
when originSize < 100000000 then 's4'
else 's5'
end as os
group by
os</pre>
```

Obtain the percentages of different results

Obtain the number and percentage of each count result for different departments. The following query statement includes subqueries and window functions. The sum(c) over() clause is used to calculate the sum of values in all rows.



Calculate the number of log entries that meet the query condition

Use the count_if clause to calculate the number of URLs that meet specified conditions and obtain the number of URLs that meet each condition by minute.

```
* |
select
count_if(uri like '%login') as login_num,
count_if(uri like '%register') as register_num,
date_format(date_trunc('minute', __time_), '%m-%d %H:%i') as time
group by
time
order by
time
limit
100
```

• You can use the following clause to calculate the number of URLs that end with login: count_if(uri like '%login')

• You can use the following clause to calculate the number of URLs that end with register: <code>count_if(uri like '%register')</code> .

4.4.13.3. Examples of time field conversion

In most cases, you need to process the time fields in log data when you query and analyze the log data. For example, you need to convert a timestamp to a specified time format. This topic provides some examples on how to convert the values of time fields.

A log may contain multiple fields that record points in time for different events. Examples:

- <u>__time__</u>: records the time when you call the API or use an SDK to write log data. You can use this field when you ship, query, and analyze log data.
 Original time field in log data: records the time when the log data is generated. This field exists in raw logs.
- Time fields in different formats are difficult to read. To simplify the read process, you can convert the time format when you query and analyze log data. Examples:
- 1. Convert the value of __time__ to a timestamp
- 2. Display the value of __time__ in a specified format
- 3. Convert the time in a log to a specified format

Convert the value of __time__ to a timestamp

You can use the from_unixtime function to convert the value of the ______ field to a timestamp.

* | select from_unixtime(__time__)

Display the value of __time__ in a specified format

To display the value of the __time__ field in the format of YYYY-MM-DD HH:MM:SS , you can use the date_format function. * | select date_format(__time__, '%Y-%m-%d %H:%i:%S')

Convert the time in a log to a specified format

To convert the value of the time field in a log to a specified format, such as YYYY-MM-DD HH:MM:SS , and then perform the GROUP BY operation on the YYYY-MM-DD part, you can use the date_format function.

Sample log

```
_topic_:
body_byte_sent: 307
hostname: www.hostl.com
http_user_agent: Mozilla/5.0 (iPhone; CPU iPhone OS 10_3_3 like Mac OS X) AppleWebKit/603.3.8 (KHTML, like Gecko) Mobile/14G60 QQ/7.1.8.452
V1_IPH_SQ_7.1.8_1_APP_A Pixe1/750 Core/UIWebView NetType/WIFI QBWebViewType/1
method: GET
refore: www.hostl.com
remote_addr: 36.63.1.23
request_length: 111
request_time: 2.705
status: 200
upstream_response_time: 0.225582883754
url: /tk0=v96
time:2017-05-17 09:45:00
```

Sample SQL statement

* | select date_format (date_parse(time,'%Y-%m-%d %H:%i:%S'), '%Y-%m-%d') as day, count(1) as uv group by day order by day asc

4.4.14. Associate Log Service with external data sources

4.4.14.1. Overview

Log Service provides the external storage feature. You can use the feature to associate Log Service with MySQL databases, Alibaba Cloud Object Storage Service (OSS) buckets, or hosted CSV files. This topic describes the scenarios, benefits, and supported external stores of the external storage feature.

Scenarios

When you analyze data, you may need to obtain different types of data from separate storage resources. For example, you need to obtain data on user operations and user behavior from Log Service, and obtain data on user properties, registration, funds, and props from a database. In this example, you must classify and analyze data and then write the analysis results to the report system of the database.

To do this, you can migrate data to a centralized storage system and then analyze the data. However, the migration process is time-consuming and labor-intensive. During migration, data must be cleansed and formatted, and network resources are consumed. To address these issues, Log Service provides API operations for external storage. You can call the API operations to achieve the following goals:

- Define mappings between data in external stores and data in Log Service. Data migration is not required.
- Use a unified query engine. You can use JOIN statements to perform JOIN queries on data in Log Service and data in external stores.
- Store query results in external stores.

Benefits

- Cost-effective
 - The external storage feature eliminates the need for data migration, which helps you reduce overall costs. Data in different storage systems is stored in different formats. The API operations that you can call to manage data also vary based on the storage systems. This results in complicated data conversion during data migration. If you use the external storage feature of Log Service, you do not need to migrate data.
- The external storage feature eliminates the need for data maintenance, which helps you reduce overall costs. If you migrate data, you must update
 and maintain the data at the earliest opportunity.
- Convenient
 - $\circ~$ You can use SQL statements to analyze data and obtain the analysis results within seconds.
- $\circ\;$ You can add charts to a dashboard and view the charts when you open the dashboard.

Supported external stores

The external storage feature of Log Service allows you to associate Log Service with MySQL databases, OSS buckets, or hosted CSV files. The following table describes the supported external stores.

Supported external store	Read from the external store	Write to the external store	Method that is used to create an external store
MySQL	Supported	Supported	API, SDK, and CLI
OSS	Supported	Supported	SQL create table
Hosted CSV files	Supported	Not supported	SDK

4.4.14.2. Associate Log Service with a MySQL database

This topic describes how to create an external store to associate Log Service with a MySQL database.

Prerequisites

- Data is collected and stored in Log Service. For more information, see Data collection.
- Data is stored in a MySQL database.

Background information

The external storage feature of Log Service allows you to associate Log Service with databases that are deployed on ApsaraDB RDS for MySQL instances, self-managed MySQL databases that are deployed on Elastic Compute Service (ECS) instances, and self-managed MySQL databases that are created in other scenarios. The external storage feature also allows you to write query and analysis results to the MySQL databases for processing. In the following descriptions, the ApsaraDB RDS for MySQL instances are referred to as RDS instances.

Procedure

- 1. Configure an allowlist for your MySQL database.
 - If you use a MySQL database that is deployed on an RDS instance, add the following CIDR blocks to the allowlist: 100.104.0.0/16, 11.194.0.0/16, and 11.201.0.0/16. For more information, see **Configure an allowlist** in **ApsaraDB RDS User Guide**.
 - If you use a self-managed MySQL database deployed on an ECS instance that resides in a virtual private cloud (VPC) and the ECS instance is added to a security group, configure security group rules to allow access from the following CIDR blocks: 100.104.0.0/16, 11.194.0.0/16, and 11.201.0.0/16. For more information, see Add security group rules in ECS User Guide.
 - If you use a self-managed MySQL database that is created in other scenarios, add the following CIDR blocks to the allowlist: 100.104.0.0/16, 11.194.0.0/16, and 11.201.0.0/16.
- 2. Create an external store
 - i. Install the Log Service CLI. For more information, see Aliyun Log Service CLI.
 - ii. Create a configuration file named /root/config.json.

User Guide-Log Service

iii. Add the following script to the /root/config.json file. Change the parameter values based on your business scenario.

{
"externalStoreName":"storename",
"storeType":"rds-vpc",
"parameter":
{
"region":"cn-qingdao-env*****",
"vpc-id":"vpc-m5eq4irc1pucp******",
"instance-id":"i-m5eeo2whsn******",
"host":"localhost",
"port":"3306",
"username":"root",
"password":"****",
"db":"scmc",
"table":"join_meta"
}

}

Parameter	Description
externalStoreName	The name of the external store. The name must be in lowercase.
storeType	The type of the data source. Set the value to rds-vpc.
region	 The region. If you use a MySQL database that is deployed on an RDS instance, set theregion parameter to the region where the RDS instance resides. If you use a self-managed MySQL database deployed on an ECS instance that resides in a VPC, set theregion parameter to the region where the ECS instance resides. If you use a self-managed MySQL database that is created in other scenarios, set theregion parameter to an empty string. Format: "region": "".
vpc-id	 The ID of the VPC. If you use a MySQL database deployed on an RDS instance that resides in a VPC, set thevpc-id parameter to the ID of the VPC where the RDS instance resides. If you use a self-managed MySQL database deployed on an ECS instance that resides in a VPC, set thevpc-id parameter to the ID of the VPC where the ECS instance resides. If you use a MySQL database deployed on an RDS instance that resides in the classic network or if you use a self-managed MySQL database that is created in other scenarios, set the vpc-id parameter to an empty string. Format: "vpc-id": "".
instance-id	 The ID of the instance. If you use a MySQL database that is deployed on an RDS instance, set theinstance-id parameter to the value of the VpcCloudInstanceId parameter that is specified for the RDS instance. If you use a self-managed MySQL database deployed on an ECS instance that resides in a VPC, set thinstance-id parameter to the ID of the ECS instance. If you use a self-managed MySQL database that is created in other scenarios, set theinstance-id parameter to an empty string. Format: "instance-id": "".
host	 The address of your MySQL database. If you use a MySQL database deployed on an RDS instance that resides in a VPC, set thehost parameter to an internal endpoint of the RDS instance. If you use a self-managed MySQL database deployed on an ECS instance that resides in a VPC, set thehost parameter to the private IP address of the ECS instance. If you use a self-managed MySQL database that is created in other scenarios, set thehost parameter to a host address of the database. Make sure that the host address is accessible.
port	 The port number. If you use a MySQL database that is deployed on an RDS instance, set theport parameter to the port of the RDS instance. If you use a self-managed MySQL database deployed on an ECS instance that resides on a VPC, set theport parameter to the MySQL service port of the ECS instance. If you use a self-managed MySQL database that is created in other scenarios, set theport parameter to the MySQL service port.
username	The username of the account that you use to log on to your MySQL database.
password	The password of the account that you use to log on to your MySQL database.
db	The name of your MySQL database.
table	The name of the table that you want to use in your MySQL database.

iv. Create an external store. Replace the value of the **project_name** parameter with the name of an actual project.

aliyunlog log create_external_store --project_name="log-rds-demo" --config="file:///root/config.json"

Related operations

• Update the MySQL external store.

aliyunlog log update_external_store --project_name="log-rds-demo" --config="file:///root/config.json"

• Delete the MySQL external store.

aliyunlog log delete_external_store --project_name="log-rds-demo" --store_name=abc

What to do next

JOIN queries on a Logstore and a MySQL database

4.4.14.3. Associate Log Service with an OSS bucket

This topic describes how to create an external store to associate Log Service with an Object Storage Service (OSS) bucket.

Prerequisites

- Logs are collected. For more information, see Data collection.
- The indexing feature is enabled and indexes are configured. For more information, see Configure indexes.
- An OSS bucket is created. For more information, see **Create buckets** in **OSS User Guide**.
- CSV files are uploaded to the OSS bucket. For more information, see Upload objects in OSS User Guide.

Benefits

The external storage feature that is used to associate Log Service with OSS buckets provides the following benefits:

- Reduced O&M workload: You can perform lightweight association analysis without the need to store all data in one storage system.
- High efficiency: You can use SQL statements to analyze data and view the analysis results within seconds. You can also create charts based on analysis results that are commonly queried. Then, you can click the charts to view the analysis results.

Procedure

- 1. Log on to the Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. Choose Log Storage > Logstores. On the Logstores tab, click the Logstore that you want to manage.
- 4. On the page that appears, enter a query statement in the search box and click Search & Analyze.
 - Execute the following SQL statement to create a virtual external table named user_meta1 and map the table to the OSS object user.csv. If the value of the result parameter in the output is true, the SQL statement is successfully executed, and an external store is created.

Define the name and table schema of the external store in the SQL statement, and define the information that is required to access OSS objects in the WITH clause. The following table describes the parameters.

Parameter	Description	
External store name	The name of the external store. The name is the same as the name of the virtual external table. Example: user_metal.	
Table schema	The properties of the virtual external table, including the column names and data types. Example: (userid bigint, nick varchar, gender varchar, province varchar, gender varchar, age bigint).	
endpoint	The internal endpoint of OSS. For more information, seeView the endpoint of a project	
accessid	The AccessKey ID of your account. For more information, seeObtain an AccessKey pair.	
accesskey	The AccessKey secret of your account. For more information, seeObtain an AccessKey pair.	
bucket	The OSS bucket in which the CSV object is stored.	
objects	The path of the CSV object. Note The value of the objects parameter is an array. The array can contain multiple elements. Each element represents an OSS object.	
type	The type of the external store. Set the value to oss .	

5. Check whether the external store is created.

Execute the following statement. If the table content that you defined is returned, the external store is created.

* | select * from user_metal

6. Perform a JOIN query on Log Service and OSS.

Execute the following statement to perform a JOIN query. A Logstore is associated with OSS objects based on the ID field in the Logstore and the userid field in the OSS objects. **test_accesslog** indicates the name of the Logstore. I indicates the alias of the Logstore. **user_meta1** indicates the name of the external store that you define. You can configure the parameters based on your business scenario.

* | select * from test_accesslog l join user_metal u on l.userid = u.userid

4.4.14.4. Associate Log Service with a hosted CSV file

Log Service allows you to upload a CSV file from your computer to Log Service by using an SDK. This way, the CSV file is hosted on Log Service and can be associated with a Logstore of Log Service. This topic describes how to perform a JOIN query on a CSV file that is hosted on Log Service and data in a Logstore of Log Service.

Prerequisites

- Logs are collected. For more information, see Data collection.
- Indexes are configured. For more information, see Configure indexes.
- A CSV file is created.
- Log Service SDK for Python is installed. For more information, see Log Service SDK for Python in Log Service Developer Guide.
- Log Service SDK for Python V0.7.3 and later are supported. You can use the pip install aliyun-log-python-sdk -U command to upgrade the SDK.

Limits

- Only one CSV file can be associated at a time.
- After you delete an external store, you cannot create an external store that has the same name as the deleted external store.
- You can associate Log Service with a CSV file that contains no more than 50 MB of data. The CSV file is uploaded to Log Service after it is compressed by using the SDK. The size of the file after compression must be less than 9.9 MB.

Sample data

The Logstore stores the logon operations of a user, and the CSV file records the basic information about the user, such as the gender and age. After you associate the Logstore with the CSV file, you can analyze the metrics for user properties.

Logstore

userid:	100001
action:	login
time_	:1637737306

CSV file

userid	nick	gender	province	age 🔍
100001	User_A	male	Liaoning	24
100002	User_B	male	Beijing	23
100003	User_C	female	Zhejiang	22
100004	User_D	female	Jiangxi	21
100005	User E	male	Guangxi	20

Procedure

- 1. Use Log Service SDK for Python to create an external store.
- For more information about Log Service SDK for Python, see Log Service SDK for Python in Log Service Developer Guide.

```
from aliyun.log import *
```

endpoint='data.cn-qingdao-env17-d01.sls-pub.inter.env17e.shuguang.com'

```
res.log_print()
```

Parameter	Description	
endpoint	The Log Service endpoint. For more information, see Obtain an endpoint in Log Service Developer Guide.	
accessKeyld	The AccessKey ID that is used to access Log Service. For more information, seeObtain an AccessKey pair. Marning We recommend that you use the AccessKey pair of a RAM user to call API operations. This prevents the AccessKey pair of your Apsara Stack tenant account from being leaked.	
accessKey	The AccessKey secret that is used to access Log Service. For more information, seeObtain an AccessKey pair.	
project	The project to which the Logstore belongs.	

ext_logstore	The name of the external store. The name is the same as the name of the virtual external table. The name must meet the following requirements: The name can contain only lowercase letters, digits, hyphens (-), and underscores (_). The name must start and end with a lowercase letter or a digit. The name must be 3 to 63 characters in length.
csv_file	The path and name of the CSV file.
Table schema	The properties of the virtual external table, including the column names and data types. The following section shows an example of a table schema. You can replace the table schema based on your business scenario. [

2. Log on to the Log Service console

3. In the Projects section, click the project that you want to manage.

4. Choose Log Storage > Logstores. On the Logstores tab, click the Logstore where logs are stored.

5. Execute the following statement to check whether the external store is created.

In the following statement, user_meta specifies the name of the external store. Replace the value with the name that you specified when you created the external store.

* | SELECT * FROM user_meta

If the content of the CSV file is returned, the external store is created.

userid ् ्	nick ‡ ्	gender	province $$= 0$	age 🌲 🍣
100001	User_A	male	Liaoning	24
100002	User_B	male	Beijing	23
100003	User_C	female	Zhejiang	22
100004	User_D	female	Jiangxi	21
100005	User_E	male	Guangxi	20

6. Execute the following statement to perform a JOIN query on the Logstore and the CSV file. The Logstore is associated with the CSV file based on the value of the **userid** field in the Logstore and the value of the **userid** field in the CSV file. website_log is the name of the Logstore. user_meta is the name of the external store that you created. You can configure the parameters based on your business scenario.

* | SELECT * FROM website_log JOIN user_meta ON website_log.userid = user_meta.userid

action	\$ 0,	userid	\$ Q,	time	\$ Q	userid 🗘 🗘	nick ‡ ୍	gender \$ ्	province $\ \ \diamondsuit \ \ \Diamond$	age
login		100003		1637738249		100003	User_C	female	Zhejiang	22
login		100004		1637738249		100004	User_D	female	Jiangxi	21
login		100005		1637738249		100005	User_E	male	Guangxi	20
login		100002		1637738249		100002	User_B	male	Beijing	23
login		100004		1637738249		100004	User_D	female	Jiangxi	21
login		100003		1637738249		100003	User_C	female	Zhejiang	22
login		100001		1637738249		100001	User_A	male	Liaoning	24

4.4.15. Visual analysis

4.4.15.1. Charts

4.4.15.1.1. Chart overview

Log Service allows you to render query and analysis results into visualized charts.

Prerequisites

Indexes are configured and the analysis feature is enabled. To enable the analysis feature, turn on Enable Analytics for the fields in the Search & Analysis panel. For more information, see Configure indexes.

? Note

• Before you configure charts, we recommend that you are familiar with the log analysis feature. For more information, see Log analysis overvi

• You must specify an analytic statement in a query statement. Log Service cannot display charts based on query results.

Usage notes

When you execute multiple query statements in sequence, the **Value Column**, **X Axis**, or **Y Axis** configurations are not automatically modified based on the current query statement. The X-axis and Y-axis configurations may remain the same as the configurations in the previous query statement. In this case, the query and analysis result of the current query statement cannot be automatically displayed on a chart. If the following errors occur, you must modify the **Properties** settings based on the current query statement:

- The dimensions that you selected are not in the query results. Check and modify the Properties settings.
- X Axis or Y Axis is unavailable. Check and modify the Properties settings.

Chart configurations

Charts are used to display query and analysis results. On the Graph tab, you can select a chart in the chart type section and configure the chart based on your business requirements.

Operation	Description
Select a chart	On the Graph tab, you can select a chart type to display the query and analysis results.
Common Settings tab	On the Common Settings tab, you can configure global settings for a chart. For example, you can select a color scheme to display the results of all query statements for the chart.
Fields tab	On the Fields tab, you can configure custom display settings for the result of a query statement or for the data of a column in the result. For example, if you select a query statement and then select a color scheme, the chart is generated based on the result of the query statement and uses the color scheme that you select.
Drill-down events	You can configure an interaction occurrence for the result of a query statement or for the data of a column in the result to analyze data from a fine-grained dimension.
Upgrade a chart	Simple Log Service provides the following versions of charts: Pro and Standard. Compared with charts (Standard), charts (Pro) provide better visualization capabilities. Simple Log Service provides the one-click migration feature to upgrade charts (Standard) to charts (Pro). After you convert a chart (Standard) to a chart (Pro), you cannot convert the chart (Pro) to a chart (Standard).

Supported chart types

- Display query results in a table
- Display query results on a line chart
- Display query results on a column chart
- Display query results on a bar chart
- Display query results on a pie chart
- Display query results on an area chart
- Display query results on a single value chart
- Display query results on a progress bar
- Display query results on a map
- Display query results in a Sankey diagram
- Display query results on a word cloud
- Display query results on a treemap chart

4.4.15.1.2. Display query results in a table

Tables are used to sort and display data for quick reference and analysis. All query results that match specified query statements can be rendered into visualized charts. By default, query results are displayed in a table.

Components

- Table header
- Row
- Column
- Where:
- The number of columns can be specified by using a SELECT statement.
- The number of rows is calculated based on the number of log entries in a specified time range. The default clause is LIMIT 100.

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project in which you want to configure a chart.
- 3. Click the name of the Logstore in which you want to configure a chart.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.
- 5. View the query result.
- By default, the query result is displayed in a table on the **Graph** tab.
- 6. On the **Properties** tab, configure the properties of the table. If you want to modify the rows and columns of the table or the entries to return on each page, you can set the parameters on the Properties tab.

Parameter D	Description
Items per Page Ti	The number of entries to return on each page.

Zebra Striping	Specifies whether to display the query result in a zebra-striped table.
Transpose Rows and Columns	Specifies whether to transpose rows and columns.
Hide Reserved Fields	Specifies whether to hide reserved fields, such astime andsource
Disable Sorting	Specifies whether to disable the sorting feature.
Disable Search	Specifies whether to disable the search feature.
Highlight Settings	If you turn on Highlight Settings, you can create rules to highlight matched rows or columns.
Sparkline	If you turn on Sparkline, you can add an area chart, a line chart, or a column chart for columns in the table.

Example

To query and analyze the distribution of page views (PVs) for different users based on status, execute the following query statement:

* |select Status, AlertDisplayName as name, COUNT(*) as count group by Status, name

4.4.15.1.3. Display query results on a line chart

This topic describes how to configure a line chart to display query results.

Background information

Line charts are used to analyze the changes of field values based on an ordered data type. In most cases, the analysis is based on a specified time range. You can use a line chart to analyze the following change characteristics of field values over a specified period of time:

- Increment or decrement
- Increment or decrement rate
- · Increment or decrement pattern, for example, periodicity
- Peak value and bottom value

You can use line charts to analyze the changes of field values over a specified period of time. You can also use line charts to analyze the changes of multiple field values in multiple lines over the same period. This way, you can analyze the relationship between different fields. For example, the values of a field are proportional or inversely proportional to the values of another field.

Each line chart consists of the following elements:

- X-axis
- Left Y-axis
- Right Y-axis (optional)
- Data point
- Line of trend changes
- Legend

Procedure

1. Log on to the Simple Log Service console

- 2. In the Projects section, click the project in which you want to configure a chart.
- 3. Click the name of the Logstore in which you want to configure a chart.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.
- 5. On the **Graph** tab, click the icon.

6. On the **Properties** tab, configure the properties of the line chart.

Parameter	Description
X Axis	The sequential data. In most cases, a time series is selected.
Left Y Axis	The numeric data. You can select one or more fields for the left Y-axis.
Right Y Axis	The numeric data. You can select one or more fields for the right Y-axis. The layer of the right Y-axis is higher than the layer of the left Y-axis.
Column Marker	The column on the left or right Y-axis. The column is displayed as a histogram.
Legend	The position of the legend in the chart. Valid values: Top, Bottom, Left, and Right.
Format Left Y-axis	The format is which the data on the left and right V axis is displayed
Format Right Y-axis	The format in which the data of the felt and right r-axis is displayed.
Line Type	The type of line that is displayed in the line chart. Valid values Straight Line and Curve .
Anomaly Point Column	The column where anomaly points are located. You can set the Anomaly Point Lower Limit and Anomaly Point Upper Limit parameters for a column. • Anomaly Point Lower Limit : Values that are less than the lower limit are highlighted in red. Anomaly Point Upper Limit Values that were all the wave differences highlighted in red.
	 Anomaly Point Upper Limit: Values that exceed the upper limit are highlighted in red.
Upper Limit Column	The area that is formed based on the values.
Lower Limit Column	
Time Series	A series of data points that are listed in chronological order.
Time Format	The time format of the time series fields.

|--|

O Note Each line in a line chart must contain more than two data points. Otherwise, the data trend cannot be generated. We recommend that you select no more than five lines for a line chart.

Example of a simple line chart

To query the page views (PVs) of the IP address 203.0.113.10 in the previous 24 hours, execute the following query statement:

remote_addr: 203.0.113.10 | select date_format(date_trunc('hour', __time_), '%m-%d %H:%i')
as time, count(1) as PV group by time order by time limit 1000

Select time for X Axis and PV for Left Y Axis. Set the Legend parameter and adjust the margins based on your business requirements.

Example of a dual Y-axis line chart

To query the PVs and number of unique visitors (UVs) in the previous 24 hours, execute the following query statement:

* | select date_format(date_trunc('hour', __time__), '%m-%d %H:%i') as time, count(1) as PV, approx_distinct(remote_addr) as UV group by time order by time limit 1000

Select time for X Axis, PV for Left Y Axis, UV for Right Y Axis, and PV for Column Marker.

4.4.15.1.4. Display query results on a column chart

This topic describes how to configure a column chart to display query results.

Background information

A column chart uses vertical or horizontal bars to show the values of different categories. You can use a column chart to count the number of values in each category.

Each column chart consists of the following elements:

- X-axis (horizontal)
- Y-axis (vertical)
- Rectangular bar
- Legend

By default, column charts in Log Service use vertical bars. Each rectangular bar has a fixed width and a variable height that indicates a value. If you select multiple columns of data for the Y-axis, a grouped column chart is used to display the data.

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project in which you want to configure a chart.
- 3. Click the name of the Logstore in which you want to configure a chart.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.
- 5. On the **Graph** tab, click the icon.

```
6. On the Properties tab, configure the properties of the column chart.
```

Note If a query statement returns no more than 20 log entries, you can use a column chart to display the query results. You can use a LIMIT clause to limit the number of rectangular bars. If the chart contains a large number of rectangular bars, the analysis results may not be displayed as expected. We recommend that you select no more than five fields for the Y-axis.

Parameter	Description
X Axis	The categorical data.
Y Axis	The numeric data. You can select one or more fields for the left Y-axis.
Legend	The position of the legend in the chart. Valid values: Top, Bottom, Left, and Right.
Format	The format in which the data on the Y-axis is displayed.
Margin	The distance between the axis and the borders of the chart. The parameters include Top Margin, Bottom Margin, Left Margin, and Right Margin.

Example of a simple column chart

To query the number of visits for each http_referer in the specified time range, execute the following query statement:

* | select http_referer, count(1) as count group by http_referer

Select http_referer for X Axis and count for Y Axis.

Example of a grouped column chart

To query the number of visits and the average bytes for each http_referer in the specified time range, execute the following query statement:

* | select http_referer, count(1) as count, avg(body_bytes_sent) as avg group by http_referer

Select http_referer for X Axis. Select count and avg for Y Axis.

4.4.15.1.5. Display query results on a bar chart

This topic describes how to configure a bar chart to display query results.

Background information

A bar chart is a horizontal column chart that is used to analyze the top N values of fields. You can configure a bar chart in a similar manner in which

you configure a column chart.

Each bar chart consists of the following elements:

- X-axis (vertical)
- Y-axis (horizontal)
- Rectangular bar
- Legend

Each rectangular bar has a fixed height and a variable width. The variable width indicates a value. If multiple columns of data are mapped to the Y-axis, you can use a grouped bar chart to display the data.

Procedure

1. Log on to the Simple Log Service console

- 2. In the Projects section, click the project in which you want to configure a chart.
- 3. Click the name of the Logstore in which you want to configure a chart.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.
- 5. On the **Graph** tab, click the _____ icon.

6. On the **Properties** tab, configure the properties of the bar chart.

? Note

- If a query statement returns no more than 20 log entries , you can use a bar chart to display the query results. You can also use a **LIMIT** clause to limit the number of rectangular bars. If the chart contains a large number of rectangular bars, analysis results may not be displayed as expected. You can also use an **ORDER BY** clause to analyze Top N values of fields. We recommend that you select no more than five fields for the Y-axis.
- You can use a grouped bar chart to display query results. However, the values represented by each rectangular bar in a group must be positively or negatively associated with each other.

Parameter	Description
X Axis	The categorical data.
Y Axis	The numeric data. You can select one or more fields for the Y-axis.
Legend	The position of the legend in the chart. Valid values: Top, Bottom, Left, and Right.
Format X-axis	The format in which the data on the X-axis is displayed.
Margin	The distance between the axis and the borders of the chart. The parameters include Top Margin, Bottom Margin, Left Margin, and Right Margin.

Example

To analyze the top 10 request URIs (request_uri) that are most frequently visited, execute the following query statement:

* | select request_uri, count(1) as count group by request_uri order by count desc limit 10

4.4.15.1.6. Display query results on a pie chart

This topic describes how to configure a pie chart to display query results.

Background information

A pie chart is used to show the percentages of different categories of data. The arc length of each segment in a pie chart is proportionate to the quantity that is represented by each category. A pie chart is divided into multiple segments based on the percentages of categories. Each segment shows the percentage of a category. The sum of all percentages is equal to 100%.

Each pie chart consists of the following elements:

- Segment
- Percentage in the text format
- Leaend

Types

Types

Log Service provides the following types of standard pie charts: pie chart, donut chart, and polar area chart.

- Donut chart
 - A donut chart is a variant of a pie chart that has a hollow center. Compared with a pie chart, a donut chart provides the following advantages:
 - Displays more information, such as the total number of occurrences of all field values.
- Allows you to compare data between two donut charts based on ring lengths. Data across different pie charts is difficult to compare.
- Polar area chart
- A polar area chart is a column chart in the polar coordinate system. Each category of data is represented by a segment with the same angle, and the radius of each segment varies based on the value. Compared with a pie chart, a polar area chart provides the following advantages:
- If a query statement returns no more than 10 log entries, you can use a pie chart to display the query results. If a query statement returns 10 to 30 log entries, you can use a polar area chart to display the query results.
- Enlarges the differences among the values of categories because the area of the segment correlates with the square of the radius. Therefore, the polar area chart is suitable for the comparison of similar values.

 A circle can be used to display a periodic pattern. Therefore, you can use a polar area chart to analyze value changes in different periods, such as weeks and months.

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project in which you want to configure a chart.
- 3. Click the name of the Logstore in which you want to configure a chart.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.
- 5. On the **Graph** tab, click the cicon.

6. On the **Properties** tab, configure the properties of the pie chart.

? Note

If a query statement returns no more than 10 log entries, you can use a pie chart or donut chart to display the query results. You can use a
 LIMIT clause to limit the number of segments. If a chart contains a large number of segments with different colors, the analysis results
 may not be displayed as expected.

• If the number of log entries exceeds 10, we recommend that you use a polar area chart or column chart.

Parameter	Description
Chart Types	The type of the chart. Valid values: Pie Chart, Donut Chart, and Polar Area Chart.
Legend Filter	The categorical data.
Value Column	The values that correspond to different categories of data.
Legend	If you turn on Show Legend , you can set this parameter to adjust the position of the legend in the chart.
Format	The format in which data is displayed.
Tick Text Format	Valid values: Percentage and Category: Percentage.
Margin	The distance between the axis and the borders of the chart. The parameters include Top Margin, Bottom Margin, Left Margin, and Right Margin.

Example of a pie chart

To analyze the percentages of the requestURI field values, execute the following query statement:

 \star | select requestURI as uri , count(1) as c group by uri limit 10



Example of a donut chart

To analyze the percentages of the requestURI field values, execute the following query statement:

 \star | select requestURI as uri , count(1) as c group by uri limit 10

Example of a polar area chart

To analyze the percentages of the requestURI field values, execute the following query statement:

 \star | select requestURI as uri , count(1) as c group by uri limit 10



4.4.15.1.7. Display query results on an area chart

This topic describes how to configure an area chart to display query results.

Background information

An area chart is built based on a line chart. The colored section between a line and the axis is an area. The color is used to highlight the trend. An area chart is similar to a line chart and shows the changes of numeric values over a specified period of time. An area chart is used to highlight the overall data trend. Both line charts and area charts display the trend and relationship between numeric values instead of specific values.

Each area chart consists of the following elements:

- X-axis (horizontal)
- Y-axis (vertical)
- Area segment

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project in which you want to configure a chart.
- 3. Click the name of the Logstore in which you want to configure a chart.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.
- 5. On the **Graph** tab, click the icon.

6. On the **Properties** tab, configure the properties of the area chart.

Note In an area chart, a single area segment must contain more than two data points. If a single area segment contains two or fewer data points, the data trend cannot be analyzed. We recommend that you select less than five area segments in an area chart.

Parameter	Description
X Axis	The sequential data. In most cases, a time series is selected.
Y Axis	The numeric data. You can select one or more fields for the Y-axis.
Legend	The position of the legend in the chart. Valid values: Top, Bottom, Left, and Right.
Format	The format in which data is displayed.
Margin	The distance between the axis and the borders of the chart. The parameters include Top Margin, Bottom Margin, Left Margin, and Right Margin.

Example of a simple area chart

To query the page views (PVs) of the IP address 203.0.113.10 in the previous 24 hours, execute the following query statement:

remote_addr: 203.0.113.10 | select date_format(date_trunc('hour', __time__), '%m-%d %H:%i') as time, count(1) as PV group by time order by time limit 1000

Select time for X Axis and PV for Y Axis.



Example of a cascade chart

To query the number of PVs and the number of unique visitors (UVs) for each hour within one day, execute the following query statement:

* | select date_format(date_trunc('hour', __time_), '%m-%d %H:%i') as time, count(1) as PV, approx_distinct(remote_addr) as UV group by time order by time limit 1000

Select time for X Axis. Select ${\tt PV}$ and ${\tt UV}$ for Y Axis.



4.4.15.1.8. Display query results on a single value chart

This topic describes how to configure a single value chart to display query results.

Background information

- A single value chart highlights a single value. Log Service supports the following types of single value charts:
- Rectangle Frame: shows a general value.
- Dial: shows the difference between the current value and a specified threshold value.
- Compare Numb Chart: shows the SQL query results of interval-valued comparison and periodicity-valued comparison functions. For more information, see Interval-valued comparison and periodicity-valued comparison functions.

By default, Rectangle Frame is selected. Rectangle Frame is the most basic type of single value chart to display data at a specified point. In most cases, this chart type is used to show the key information at a specified point in time. To display a proportional metric, you can select Dial.

Each single value chart consists of the following elements:

Numeric value

- Unit (optional)
- Description (optional)
- Chart type

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project in which you want to configure a chart.
- 3. Click the name of the Logstore in which you want to configure a chart.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.
- 5. On the **Graph** tab, click the 123 icon.

6. On the **Properties** tab, configure the properties of the single value chart.

(2) Note Log Service normalizes data in charts that contain numeric values. For example, 230000 is processed as 230K . You can include mathematical calculation functions in query statements to customize numeric formats. For more information, see Mathematical calculation functions.

• The following table describes the parameters of a rectangle frame.

Parameter	Description
Chart Types	The type of the single value chart. If you select Rectangle Frame , query results are displayed in a rectangle frame.
Value Column	The value that is displayed in the chart. By default, the data in the first row of the specified column is displayed.
Unit	The unit of the data.
Unit Font Size	The font size of the unit. You can drag the slider to adjust the font size. Valid values: 10 to 100. Unit: pixels.
Description	The description of the value.
Description Font Size	The font size of the description. You can drag the slider to adjust the font size. Valid values: 10 to 100. Unit: pixels.
Format	The format in which data is displayed.
Font Size	The font size of the value. You can drag the slider to adjust the font size. Valid values: 10 to 100. Unit: pixels.
Font Color	The color of the value, unit, and description in the chart. You can use the default color or select a color.
Background Color	The color of the background. You can use the default color or select a color.

$\circ\;$ The following table describes the parameters of a dial.

Parameter	Description
Chart Types	The type of the single value chart. If you select Dial , query results are displayed on a dial.
Actual Value	The value that is displayed in the chart. By default, the data in the first row of the specified column is displayed.
Unit	The unit of the value on the dial.
Font Size	The font size of the value and unit. Valid values: 10 to 100. Unit: pixels.
Description	The description of the value.
Description Font Size	The font size of the description. You can drag the slider to adjust the font size. Valid values: 10 to 100. Unit: pixels.
Dial Maximum	The maximum value of the scale on the dial. Default value: 100.
Use Query Results	If you turn on Use Query Results, Dial Maximum is replaced by Maximum Value Column. Then, you can select the maximum value from query results for this parameter.
Format	The format in which data is displayed.
Colored Regions	The number of segments that divide the dial. Each segment is displayed in a different color. Valid values: 2, 3, 4, and 5. Default value: 3.
Region Max Value	The maximum value of the scale in each colored segment of the dial. By default, the maximum value in the last segment is the maximum value on the dial. You do not need to specify this value.
Font Color	The color of the value on the dial.
Region	The colored segments that divide the dial. By default, a dial is evenly divided into three segments. The segments are displayed in blue, yellow, and red. If you set Colored Regions to a value greater than 3, the added segments are displayed in blue by default. You can change the color of each segment.

• The following table describes the parameters of a compare numb chart.

Parameter	Description
Chart Types	The type of the single value chart. If you select Compare Numb Chart, query results are displayed on a compare numb chart.
Show Value	The value that is displayed in the center of the compare numb chart. In most cases, this value is set to the statistical result that is calculated by the related comparison function in the specified time range.
Compare Value	The value that is compared with the threshold. In most cases, this value is set to the result of the comparison between the statistical results that are calculated by the related comparison function in the specified time range and in the previously specified time range.
Font Size	The font size of the show value. Valid values: 10 to 100. Unit: pixels.
Unit	The unit of the show value.
Unit Font Size	The font size of the unit for the show value. Valid values: 10 to 100. Unit: pixels.
Compare Unit	The unit of the compare value.
Compare Font Size	The font size of the compare value and unit. Valid values: 10 to 100. Unit: pixels.
Description	The description of the show value and its growth trend.
Description Font Size	The font size of the description. Valid values: 10 to 100. Unit: pixels.
Trend Comparison Threshold	 The value that is used to measure the variation trend of the compare value. For example, the compare value is -1. If you set Trend Comparison Threshold to 0, a down arrow that indicates a value decrease is displayed on the page. If you set Trend Comparison Threshold to -1, the value remains unchanged. The system does not display the trend on the page. If you set Trend Comparison Threshold to -2, an up arrow that indicates a value increase is displayed on the page.
Format	The format in which data is displayed.
Font Color	The color of the show value and its description.
Growth Font Color	The font color of the compare value that is greater than the threshold.
Growth Background Color	The background color that is displayed when the compare value is greater than the threshold.
Decrease Font Color	The font color that is displayed when the compare value is less than the threshold.
Decrease Background Color	The background color that is displayed when the compare value is less than the threshold.
Equal Background Color	The background color that is displayed when the compare value is equal to the threshold.

Examples

To view the number of page views (PVs), execute the following query statements. The analysis results are displayed in charts.

Rectangle frame

To view the number of PVs in the previous 15 minutes, execute the following query statement:

* | select count(1) as pv

- Dial
- To view the number of PVs in the previous 15 minutes, execute the following query statement:

* | select count(1) as pv

* | select diff[1],diff[2], diff[1]-diff[2] from (select compare(pv , 86400) as diff from (select count(1) as pv from log))

4.4.15.1.9. Display query results on a progress bar

This topic describes how to configure a progress bar to display query results.

Background information

A progress bar shows the percentage of the actual value of a field to the maximum value of the field. You can configure the properties of a progress bar to change the style and configure display rules for the progress bar.

Each progress bar consists of the following elements:

- Actual value
- Unit (optional)
- Total value

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project in which you want to configure a chart.
- 3. Click the name of the Logstore in which you want to configure a chart.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.
- 5. On the **Graph** tab, click the _____ icon.

6. On the **Properties** tab, configure the properties of the progress bar.

Compare numb chart

To view and compare the PVs on the current day and in the previous day, execute the following query statement:

Parameter	Description
Actual Value	The actual value in the chart. By default, data in the first row of the specified column is displayed.
Unit	The unit of the value in the progress bar.
Total Value	The maximum value indicated by the progress bar. Default value: 100.
Maximum Value Column	The maximum value in the specified column. If you turn on Use Query Results , Total Value is replaced by Maximum Value Column . Then, you can select the maximum value from the query results for this parameter.
Use Query Results	If you turn on Use Query Results, Total Value is replaced by Maximum Value Column. Then, you can select the maximum value from the query results for this parameter.
Edge Shape	The edge shape of the progress bar.
Vertical Display	Specifies whether to display the progress bar in vertical display mode.
Font Size	The font size of the value in the progress bar.
Thickness	The thickness of the progress bar.
Background Color	The background color of the progress bar.
Font color	The font color of the value in the progress bar.
Default Color	The default color of the progress bar.
Color Display Mode	The display mode of the progress bar.
Start Color	The start color of the progress bar. This parameter is available if you select Gradient for Color Display Mode.
End Color	The end color of the progress bar. This parameter is available if you select Gradient for Color Display Mode .
Display Color	The display color of the progress bar. This parameter is available if you select Display by Rule for Color Display Mode .
Operator	The condition that is used to determine whether to display the progress bar in the color specified by Display Color. This parameter is available if you select Display by Rule for Color Display Mode .
Threshold	The threshold based on which the color of the progress bar is determined. This parameter is available if you select Display by Rule for Color Display Mode .

Example

To calculate the ratio of the page views (PVs) of the current hour to the PVs of the same period of time on the previous day, execute the following query statement:

* | SELECT diff[1] AS today, diff[2] AS yesterday, diff[3] AS ratio FROM (SELECT compare(PV,86400) AS diff FROM (SELECT count(*) AS PV FROM log))

4.4.15.1.10. Display query results on a map

This topic describes how to configure a map to display query results.

Background information

You can color and mark a map to display geographic data. Log Service provides the map of China. The display modes of an AMap include the anchor point and heat map. You can use specific functions in query statements to display analysis results as maps.

Each map consists of the following elements:

- Map canvas
- Colored area

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project in which you want to configure a chart.
- 3. Click the name of the Logstore in which you want to configure a chart.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analytics.
- 5. On the **Graph** tab, use the map of China, click the right icon..
- On the **Properties** tab, configure the properties of the map. In this example, the map of China is selected.

Parameter	Description
Provinces	The location information that is recorded in logs. The information is displayed in one of the following dimensions based on the map type:

Cloud Defined Storage

Value Column	The amount of data at the location.
Show Legend	If you turn on Show Legend, the legend information is displayed.

Example of a map of China

To display query results on a map of China, you can execute the following query statement in which the ip_to_province function is used:

* | select ip_to_province(remote_addr) as address, count(1) as count group by address order by count desc limit 10

Select address for Provinces and count for Value Column.

4.4.15.1.11. Display query results on a flow chart

This topic describes how to configure a flow chart to display query results.

Background information

A flow chart, also known as ThemeRiver, is a stacked area chart around a central axis. The banded branches with different colors indicate different categorical data. The width of the band indicates the numeric value. By default, the time information of the data is mapped to the X-axis. A flow chart can display the data in three dimensions.

You can select Line Chart or Column Chart for the Chart Types parameter. If you select Column Chart, a stacked column chart is displayed. In a stacked column chart, each category of data starts from the top of the last column of categorical data.

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project in which you want to configure a chart.
- 3. Click the name of the Logstore in which you want to configure a chart.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.
- 5. On the **Graph** tab, click the cicon.

6. On the **Properties** tab, configure the properties of the flow chart.

Parameter	Description
Chart Types	The type of the chart. Valid values: Line Chart, Area Chart, and Column Chart. Default value: Line Chart.
X Axis	The sequential data. In most cases, a time series is selected.
Y Axis	The numeric data. You can select one or more fields for the Y-axis.
Aggregate Column	The field information that must be aggregated as the third point for comparison.
Legend	The position of the legend in the chart. Valid values: Top, Bottom, Left, and Right.
Format	The format in which data is displayed.
Margin	The distance between the axis and the borders of the chart. The parameters include Top Margin, Bottom Margin, Right Margin, and Left Margin.

Example

A flow chart is suitable for displaying data in three dimensions, for example, time, category, and numeric value. In this example, the following query statement is executed:

* | select date_format(from_unixtime(__time__ - __time__% 60), '%H:%i:%S') as minute, count(1) as c, request_method group by minute, request_method order by minute asc limit 100000

Select minute for X Axis, c for Y Axis, and request_method for Aggregate Column.



4.4.15.1.12. Display query results in a Sankey diagram

This topic describes how to configure a Sankey diagram to display query results.

Background information

A Sankey diagram is a type of flow chart. A Sankey diagram shows the flow from one set of values to another set of values. You can use Sankey diagrams to collect statistics about network traffic flows. A Sankey diagram contains the values of the source , target , and value fields. The source and target fields describe the source and target nodes, and the value field describes the flows from the source node to the target node.

Each Sankey diagram consists of the following elements:

- Node
- Edge
- A Sankey diagram has the following features:
- The start flow is equal to the end flow. The sum of the widths of all main edges is equal to the sum of the widths of all branch edges. This allows you to manage and maintain a balanced flow of all traffic.
- The edge width in a row represents the volume of traffic in a specific status.

• The width of an edge between two nodes represents the flow volume in a status.

The following table describes the data that can be	displayed in a Sankey diagram.

source	target	value
nodel	node2	14
nodel	node3	12
node3	node4	5

The following figure shows the data relationships in a Sankey diagram.



Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project in which you want to configure a chart.
- 3. Click the name of the Logstore in which you want to configure a chart.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.
- 5. On the **Graph** tab, click the icon.

6. On the Properties tab, configure the properties of the Sankey diagram.

Parameter	Description
Start Column	The start node.
End Column	The end node.
Value Column	The volume of traffic between the start node and end node.
Margin	The distance between an axis and the borders of the chart. The parameters include Top Margin, Bottom Margin, Right Margin , and Left Margin .

Example

If a log contains the source , target , and value fields, you can use a nested subquery to obtain the sum of all steamValue values.

* | select sourceValue, targetValue, sum(streamValue) as streamValue from (select sourceValue, targetValue,

streamValue, __time__ from log group by sourceValue, targetValue, streamValue, __time__ order by __time__ desc) group by sourceValue, targetValue

4.4.15.1.13. Display query results on a word cloud

This topic describes how to configure a word cloud to display query results.

Background information

A word cloud shows text data. A word cloud is a cloud-like and colored image composed of words. You can use a word cloud to display a large amount of text data. The font size or color of a word indicates the significance of the word. This allows you to identify whether a word is significant in an efficient manner.

The words in a word cloud are sorted.

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project in which you want to configure a chart.
- 3. Click the name of the Logstore in which you want to configure a chart.
- 4. Enter a query statement in the search box, specify a time range, and then click **Search & Analyze**.
- 5. On the **Graph** tab, click the icon.

6. On the Properties tab, configure the properties of the word cloud.

Parameter	Description
Word Column	The words that you want to display.
Value Column	The numeric value that corresponds to a word.

	The font size of a word.
Font Size	 The minimum font size ranges from 10 pixels to 24 pixels. The maximum font size ranges from 50 pixels to 80 pixels.

Example

To query the distribution of hostnames in NGINX logs, execute the following query statement:

 \star | select hostname, count(1) as count group by hostname order by count desc limit 1000

Select hostname for Word Column and count for Value Column.



4.4.15.1.14. Display query results on a treemap chart

This topic describes how to configure a treemap chart to display query results.

Background information

A treemap chart consists of multiple rectangles that represent the data volumes. A larger rectangle area represents a larger proportion of the categorical data.

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project in which you want to configure a chart.
- 3. Click the name of the Logstore in which you want to configure a chart.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.
- 5. On the **Graph** tab, click the icon.

6. On the **Properties** tab, configure the properties of the treemap chart.

Parameter	Description
Legend Filter	The field that includes categorical data.
Value Column	The numeric value of a field. A greater field value represents a larger rectangle.

Example

To query the distribution of hostnames in NGINX logs, execute the following query statement:

* | select host, count(1) as count group by host order by count desc limit 1000

Select host for Legend Filter and select count for Value Column.

			www.sk.mock
	www.s		
www.qb.mock.com			w.qi.mock.com

4.4.15.2. Charts (Pro)

4.4.15.2.1. Overview of charts (Pro)

Log Service can render query and analysis results into charts. You can use a chart (Pro) to display the results of multiple query statements. You can also configure personalized display settings for the results of each query statement.

Usage notes

- Before you can use charts (Pro), you must configure indexes and enable the analysis feature. You can turn on the switches in the **Enable Analytics** column for the fields that you want to analyze in the **Search & Analysis** panel. For more information, see Configure indexes.
- Log Service can display the query and analysis results in charts only after query statements are executed. A query statement is in the following format: Search statement|Analytic statement.

Chart types

- Log Service provides the following types of charts (Pro):
- Table (Pro)
- Line chart (Pro)
- Flow chart (Pro)

- Column chart (Pro)
- Single value chart (Pro)
- Pie chart (Pro)

Operations

Charts are used to display query and analysis results. You can select different types of charts from the chart type section and configure settings for the charts.

Operation	Description
Select a chart	Select a chart type to display your query and analysis results.
Common Settings tab	Configure global settings for a chart. For example, you can select a color scheme to display the results of all query statements for the chart.
Fields tab	Configure personalized display settings for the results of a single query statement or for a single column of data in the results. For example, if you select a query statement and then select a color scheme, the chart is generated based on the results of the query statement and uses the color scheme that you select.
Drill-down eventsDrill-down events	Configure a drill-down event for the results of a single query statement or for a single column of data in the results to analyze data from a finer-grained dimension. Drill-down events include events to open a Logstore, open quick analysis, open a dashboard, open trace analysis, open trace details, and customize an HTTP link.
Add a chart (Pro) to a dashboard	Add a chart to a dashboard.

4.4.15.2.2. Add a chart (Pro) to a dashboard

Log Service allows you to save query and analysis results as charts to a dashboard. This topic describes how to add a chart (Pro) to a dashboard.

Prerequisites

- Data is collected. For more information, see Data collection overview.
- If log data is collected, indexes are configured for the log data. For more information, see Configure indexes.
- A dashboard is created. For more information, see Create a dashboard.

Limits

You can add up to 200 charts to a dashboard.

Entry point

You can add a chart on a Logstore page or a dashboard page. The configuration items remain unchanged regardless of the entry point that you use. • Logstore page

If you want to add a chart on a Logstore page, you must execute a query statement, select a chart, and then save the chart to a dashboard. For more information about how to access a Logstore page, see Query and analyze logs.

Dashboard page

If you want to add a chart on a dashboard page, you must select a dashboard, select a chart, and then execute a query statement. For more information about how to access a dashboard page, see Create a dashboard.

4.4.15.2.3. Attributes of charts (Pro)

Charts (Pro) allow you to configure global settings for charts and configure personalized display settings for the results of a single query statement or for a single column of data in the results.

Common Settings tab

On the **Common Settings** tab, you can configure global settings for a chart. The settings take effect on the chart that is generated based on the results of all query statements.

The global settings include the basic settings, standard settings, and attributes for a chart. For more information, see the following topics:

- Table (Pro)
- Line chart (Pro)
- Flow chart (Pro)
- Column chart (Pro)
- Single value chart (Pro)
- Pie chart (Pro)

Common Settings	Fields	Interaction Occ 🧹	>
 Chart Types 			
 Basic Configuration 	ons		
 Standard Configu 	urations		
v Search & Analysi	s Settings		
 Table Configuration 	ions		
 Field/Column Set 	tings		
 Threshold 			
 Replace Variable 			
✓ Mapping Value			
 Documentation 			

Fields tab

On the **Fields** tab, you can configure personalized display settings for the results of a single query statement or for a single column of data in the results. The settings are chart attributes and take effect only on the chart that is generated based on the results of the selected query statement or based on a single column of data in the results.

For more information about the descriptions and examples of parameter configurations on the Fields tab, see the following topics:

- Table (Pro)
- Line chart (Pro)
- Flow chart (Pro)
- Column chart (Pro)
- Single value chart (Pro)
- Pie chart (Pro)



Differences between the Common Settings and Fields tabs

The two tabs provide similar configuration items but differ in the applicable scopes of the configuration items. Settings on the Common Settings tab take effect on a chart that is generated based on the results of all query statements. Settings on the Fields tab take effect on a chart that is generated based on the results of a single query statement or on a single column of data in the results.

For example, you want to display the numbers of page views (PVs) for two websites within the current hour in a line chart and you want to distinguish between the numbers by color and legend. If you configure **Display Name** and **Color Scheme** on the **Common Settings** tab, the line segments for the two websites have the same color and legend name. As a result, you cannot distinguish between the numbers. In this case, you can configure personalized display settings on the Fields tab to distinguish between the numbers.

Common Settings tab

- If you set **Color Scheme** to **Solid** and **Display Name** to **PV**, the colors and legend names for the two line segments are the same.
- Fields tab
 - You can add the A > pv field to configure settings for the line segment that represents the PV data in the query and analysis results of Website A.
 You can select the red color for Standard Configurations > Color Scheme and enter PV of Website A for Standard Configurations > Display Name.
- You can add the B > pv field to configure settings for the line segment that represents the PV data in the query and analysis results of Website B.
 You can select the yellow color for Standard Configurations > Color Scheme and enter PV of Website B for Standard Configurations > Display Name.

4.4.15.2.4. Drill-down events

When you create a chart (Pro) in a dashboard, you can configure a drill-down event. This topic describes the drill-down events that you can configure.

Introduction

Drill-down events are important for data analysis. You can use drill-down events to switch between the levels of data dimensions and the analysis granularities to obtain more detailed information. If you use a chart (Pro), you can configure a drill-down event for the results of a single query statement or for a single column of data in the results. Drill-down events include the events to open a Logstore, open a saved search, open a dashboard, open trace analysis, open trace details, and customize an HTTP link.

For more information about how to configure a drill-down event, see Add a chart (Pro) to a dashboard.

Open Logstore

You can select Open Logstore from the Add Event drop-down list to configure a drill-down event. When the drill-down event is triggered, you are navigated to the page of the Logstore that you specify.

• Prerequisites

A Logstore is created. For more information, see Create a Logstore.

Parameters

Parameter	Description
Custom Name	The name of the drill-down event.
Select a project	The project to which the Logstore belongs.
Select Logstore	The Logstore that is created. When the drill-down event is triggered, you are navigated to the page of the Logstore.
Time Range	 The time range of the data to query in the Logstore. You can use one of the following time ranges: Default time range: After you click a value in the chart that you create and you are navigated to the page of the Logstore, the time range on the page of the Logstore is the default time range, which is 15 Minutes(Relative). Chart time range: After you click a value in the chart that you create and you are navigated to the page of the Logstore, the time range on the page of the Logstore is the time range of the chart when the drill-down event is triggered. Relative time range: After you click a value in the chart that you create and you are navigated to the page of the Logstore, the time range on the page of the Logstore is the time range of the chart when the drill-down event is triggered. Relative time range: After you click a value in the chart that you create and you are navigated to the page of the Logstore, the time range on the page of the Logstore is the relative time range that you specify for the Time Range parameter. Time frame: After you click a value in the chart that you create and you are navigated to the page of the Logstore, the time range on the page of the Logstore is the relative time range that you specify for the Time Range parameter.
Inherit Filtering Conditions	If you turn on Inherit Filtering Conditions , the filter conditions that are added to your dashboard are synchronized to the Logstore. Then, you can enter a query statement after the filter conditions. The query statement and the filter conditions are evaluated by using the AND operator.
Filter Statement	The filter statement, which can be synchronized to the Logstore. You can enter a query statement after the filter statement. The two statements are evaluated by using the AND operator. You can add variables to the filter statement. After you add a variable, the value of the variable is used as a filter condition.

Open Saved Search

v ou can select Open Saved Search from the Add Event drop-down list to configure a drill-down event. When the drill-down event is triggered, you are navigated to the page of the saved search that you specify.

Prerequisites

A saved search is created. For more information, see Saved search.

If you want to configure a variable for the drill-down event, ensure that the variable is configured in the query statement of the saved search. For more information, see Variables.

• Parameters

Parameter	Description
Custom Name	The name of the drill-down event.
Select a project	The project to which the saved search belongs.
Select Saved Search	The saved search that is created. When the drill-down event is triggered, you are navigated to the page of the saved search.
Time Range	 The time range of the data to query on the page of the saved search. You can use one of the following time ranges: Default time range: After you click a value in the chart that you create and you are navigated to the page of the saved search, the time range on the page of the saved search is the default time range, which is 15 Minutes(Relative). Chart time range: After you click a value in the chart that you create and you are navigated to the page of the saved search, the time range of the saved search is the default time range, which is 15 Minutes(Relative). Chart time range: After you click a value in the chart that you create and you are navigated to the page of the saved search, the time range of the chart when the drill-down event is triggered. Relative time range: After you click a value in the chart that you create and you are navigated to the page of the saved search, the time range on the page of the saved search is the relative time range that you specify for the Time Range parameter. Time frame: After you click a value in the chart that you create and you are navigated to the page of the saved search, the time range on the page of the saved search is the relative time range that you specify for the Saved search, the time range on the page of the saved search is the time frame that you specify for the Time Range parameter.
Inherit Filtering Conditions	If you turn on Inherit Filtering Conditions , the filter conditions that are added to your dashboard are synchronized to the saved search. Then, you can enter a query statement after the filter conditions. The query statement and the filter conditions are evaluated by using the AND operator.
Filter Statement	The filter statement, which can be synchronized to the saved search. You can enter a query statement after the filter statement. The two statements are evaluated by using the AND operator. You can add variables to the filter statement. After you add a variable, the value of the variable is used as a filter condition.
Dynamic Variables	 Log Service allows you to modify the query statement in the saved search by using variables. If the variable that you specify for Dynamic Variables is the same as the variable in the query statement of the saved search, the variable in the query statement is replaced with the value of the variable that you specify for Dynamic Variables. Variable: the name of the variable. Column in which the variable resides: The variable values in this column are used to dynamically replace the variable in the query statement. Note You can add up to five dynamic variables.

	 Value: The variable values in this column are fixed values and are used to replace the variable in the query statement. 	Static Variables	Log Service allows you to modify the query statement in the saved search by using variables. If the variable that you specify for Static Variables is the same as the variable in the query statement of the saved search, the variable in the query statement is replaced with the value of the variable that you specify for Static Variables. • Variable: the name of the variable.
--	---	------------------	---

Open Dashboard

You can select Open Dashboard from the Add Event drop-down list to configure a drill-down event. When the drill-down event is triggered, you are navigated to the page of the dashboard that you specify.

Prerequisites

A dashboard is created and a chart is created in the dashboard. For more information, see Add a chart (Pro) to a dashboard.

If you want to configure a variable for the drill-down event, ensure that the variable in the query statement is configured for the chart that you create in the dashboard. For more information, see Variables.

• Parameters

Parameter	Description
Custom Name	The name of the drill-down event.
Select a project	The project to which the dashboard belongs.
Select Dashboard	The dashboard that is created. When the drill-down event is triggered, you are navigated to the page of the dashboard.
Time Range	 The time range of the data to query in the dashboard. You can use one of the following time ranges: Default time range: After you click a value in the chart that you create and you are navigated to the page of the dashboard, the time range on the page of the dashboard is the default time range, which is 15 Minutes(Relative). Chart time range: After you click a value in the chart that you create and you are navigated to the page of the dashboard, the time range on the page of the dashboard is the default time range, which is 15 Minutes(Relative). Chart time range: After you click a value in the chart that you create and you are navigated to the page of the dashboard, the time range of the dashboard is the time range of the chart when the drill-down event is triggered. Relative time range: After you click a value in the chart that you create and you are navigated to the page of the dashboard is the time range of the relative time range that you specify for the Time Range parameter. Time frame: After you click a value in the chart that you create and you are navigated to the page of the dashboard, the time range on the page of the chart wou create and you are navigated to the page of the dashboard, the time range on the page of the chart that you create and you are navigated to the page of the dashboard, the time range on the page of the dashboard is the time frame that you specify for the Time Range parameter.
Inherit Filtering Conditions	If you turn on Inherit Filtering Conditions, the filter conditions that are added to your dashboard are synchronized to the created dashboard.
Inherit Variables	If you turn on Inherit Variables , the variables that are configured for your dashboard are synchronized to the created dashboard.
Filter Statement	The filter statement, which can be synchronized to the dashboard. You can add variables to the filter statement. After you add a variable, the value of the variable is used as a filter condition.
Dynamic Variables	Log Service allows you to synchronize the variable that you specify to the dashboard. • Variable: the name of the variable. • Column in which the variable value resides: The variable values in this column are dynamically synchronized to the dashboard. ⑦ Note You can add up to five dynamic variables.
Static Variables	Log Service allows you to synchronize the variable that you specify to the dashboard. • Variable: the name of the variable. • Value: The variable values in this column are fixed values and are synchronized to the dashboard. ⑦ Note You can add up to five static variables.

Create Custom HTTP URL

You can select Create Custom HTTP URL from the Add Event drop-down list to configure a drill-down event. When the drill-down event is triggered, you are navigated to the page that you specify.

- Prerequisites
- An HTTP URL is available.
- Parameters

Parameter	Description
Custom Name	The name of the drill-down event.
Protocol	The protocol for the URL.
URL	The URL of the page to which you want to be navigated.
Transcoding	If you turn on Transcoding , the HTTP URL is encoded.

Open Trace Analysis

You can select Open Trace Analysis from the Add Event drop-down list to configure a drill-down event. When the drill-down event is triggered, you are navigated to the analysis page of the trace instance that you specify.

Prerequisites

A trace instance is created and traces are collected to Log Service. For more information, see Overview.

• Parameters

Parameter	Description
Custom Name	The name of the drill-down event.
Time Range	 The time range of the data to query on the analysis page of the trace instance that is created. You can use one of the following time ranges: Default time range: After you click a value in the chart that you create and you are navigated to the analysis page of the trace instance, the time range on the analysis page of the trace instance is the default time range, which is 15 Minutes(Relative). Chart time range: After you click a value in the chart that you create and you are navigated to the analysis page of the trace instance, the time range on the analysis page of the trace instance is the default time range, which is 15 Minutes(Relative). Chart time range: After you click a value in the chart that you create and you are navigated to the analysis page of the trace instance, the time range on the analysis page of the trace instance, the time range on the analysis page of the trace instance, the time range on the analysis page of the trace instance, the time range on the analysis page of the trace instance is the relative time range that you specify for the Time Range parameter. Time frame: After you click a value in the chart that you create and you are navigated to the analysis page of the trace instance, the time range on the analysis page of the trace instance is the relative time range that you specify for the Time Range parameter.
Trace Instance	The trace instance that is created. When the drill-down event is triggered, you are navigated to the analysis page of the trace instance.
Filter Statement	The filter statement, which can be synchronized to the analysis page of the trace instance. You can add variables to the filter statement. After you add a variable, the value of the variable is used as a filter condition.

Open Trace Details

You can select Open Trace Details from the Add Event drop-down list to configure a drill-down event. When the drill-down event is triggered, you are navigated to the details page of the trace instance that you specify.

Prerequisites

A trace instance is created and traces are collected to Log Service. For more information, see Overview.

• Parameters

Parameter	Description
Custom Name	The name of the drill-down event.
Time Range	 The time range of the data to query on the details page of the trace instance that is created. You can use one of the following time ranges: Default time range: After you click a value in the chart that you create and you are navigated to the details page of the trace instance, the time range on the details page of the trace instance is the default time range, which is 15 Minutes(Relative). Chart time range: After you click a value in the chart that you create and you are navigated to the details page of the trace instance, the time range on the details page of the trace instance, its time range on the details page of the trace instance. The time range on the details page of the trace instance is the time range of the chart when the drill-down event is triggered. Relative time range: After you click a value in the chart that you create and you are navigated to the details page of the trace instance, the time range on the details page of the trace instance, is the range on the details page of the trace instance is the relative time range that you specify for the Time Range parameter. Time frame: After you click a value in the chart that you create and you are navigated to the details page of the trace instance, the time range on the details page of the trace instance is the relative time range that you specify for the Time Range parameter.
Trace Instance	The trace instance that is created. When the drill-down event is triggered, you are navigated to the details page of the trace instance.
Trace ID	The ID of the trace instance.
Span ID	The ID of the span.

Configuration example

This section provides an example on how to configure a drill-down event by using a Logstore named website_log that stores NGINX access logs and two dashboards named RequestMethod and destination_drilldown.

- Add a table of request methods to the RequestMethod dashboard and configure a drill-down event for the table to open the destination_drilldown dashboard.
- Add a line chart that displays the trend of page views (PVs) over time to the destination_drilldown dashboard.

After the configurations are complete, you can click a request method on the RequestMethod dashboard to access the destination_drilldown dashboard. Then, you can view the trend of PVs over time on the destination_drilldown dashboard. You can perform the following operations to configure the drilldown event:

- 1. Add a line chart that displays the trend of PVs over time to the destination_drilldown dashboard.
- Enter the following query statement. \${{method | PUT}} specifies a variable. For more information, see Add a chart (Pro) to a dashboard.

request_method: \${{method|PUT}} | SELECT __time__ - _time__ %60 AS time, COUNT(1) AS PV GROUP BY time ORDER BY time

2. Add a table of request methods to the RequestMethod dashboard and configure a drill-down event for the table.

- Enter the following query statement. For more information, see Add a chart (Pro) to a dashboard.
- You can configure the drill-down event based on the following configuration.
- Select destination_drilldown for Select Dashboard.
- Specify the **method** variable and select the **request_method** column for Dynamic Variables.
- 3. Verify the drill-down event.

On the RequestMethod dashboard, click **GET** and click **Open Dashboard**. You are navigated to the destination_drilldown dashboard. The trend of PVs for GET requests over time is displayed in a line chart.

4.4.15.2.5. Variables

Variables are placeholders for values. You can use variables in query statements. You can use variables to create dashboards that are more interactive and dynamic.

Entry point

(2) Note If you want to execute a query statement on the query and analysis page of a Logstore, you cannot specify variables in the query statement.

- 1. Log on to the Log Service console.
- 2. Access the Dashboard page.
- i. In the Projects section, click the project that you want to manage.
- ii. In the left navigation sidebar, click Dashboard.
- iii. In the Dashboard list, click the dashboard that you want to manage.
- 3. On the dashboard page that appears, click Edit.
- 4. Click Create Chart or find the chart that you want to manage and choose . > Edit to go to the edit page.
- 5. Specify variables in the query statement.

A (?)	Logstore (SQL)	\sim	website_log	~	I Ú
1	request_method: § ORDER BY time	{{method PU	T}} SELECT	timetime %60 AS time, COUNT(1) AS PV GROUP BY time	

Configuration description

If you specify a variable in a drill-down event or a filter, you must add the variable to the query statement. The variable is in the
s{{Variable
name|Default value}}
format. For example, you can specify host=~"\${{host|^.*}}" for host=~"^.*".

Drill-down events

For example, you can specify a variable in the query statement of Chart A and add Chart A to Dashboard A. You can also configure a drill-down event to open Dashboard A for Chart B and specify the variable in the drill-down event. When you click a value in Chart B to trigger the drill-down event, the system replaces the variable in the query statement of Chart A with the value in Chart B and re-executes the query statement. Then, Chart A is updated. For more information, see Drill-down events.

Filters

If you add a filter of the Replace Variable type on a dashboard and the dashboard contains a chart whose query statement includes the variable specified for the filter, the system automatically replaces the variable in the query statement for the chart with the value that you select for the filter. The variable configuration applies to all charts whose query statements include the variable on the dashboard. For more information, see Add a filter.

Variable replacement

This method adds a filter of the Replace Variable type to a single chart. After you configure the settings of variable replacement on the **Common Settings** tab, Log Service adds a filter in the upper-left corner of the chart. You can select a value from the filter drop-down list. After you select a value, Log Service automatically replaces the variable in the query statement of the chart with the variable value indicated by the value that you select, and performs a query and analysis operation. For more information, see Example 2: Configure variable replacement.

Configuration examples

Example 1: Specify variables in a query statement

In the following query statement, change the value of the **request_method** field to the <code>\${{method}PUT}}</code> variable:

request_method: * | SELECT __time__ - __time__ %60 AS time, COUNT(1) AS PV GROUP BY time ORDER BY time

A (?)	Logstore (SQL)	\sim	website_log	\checkmark	E) @
1	request_method: \$- DRDER BY time	{method PU	T}} SELECTtime	*60 AS time, COUNT	(1) AS PV GROUP BY time

Example 2: Configure variable replacement

1. In the following query statement, change the value **60** to the \${{date | 60}} variable:

* | select __time__ - __time__ % 60 as time, COUNT(*) as pv, avg(request_time) as duration, request_method GROUP BY time, request_method ord er by time limit 1000

2. Configure the settings of variable replacement on the Common Settings tab.

Set Variable Key to date and Display Name below Variable Key to time. Set Display Name below Variable Values to min and hour, and set Replaced Value to 60 for min and 3600 for hour.

4.4.15.2.6. Table (Pro)

You can use a table (Pro) to visualize the results of multiple query statements. You can also configure personalized display settings on the Fields tab. This topic describes the basic configurations of a table (Pro).

Introduction

Tables are used to sort and display data for quick reference and analysis. By default, the results of query statements are displayed in tables. For more information about how to create a table, see Add a chart (Pro) to a dashboard.

Configurations on the Common Settings tab

You can configure global settings for a table on the Common Settings tab.

• Parameters in the Basic Configurations section

Parameter	Description
Title	The title of the table.
Show Title	If you turn on Show Title , the title of the table is displayed.
Show Border	If you turn on Show Border , the borders of the table are displayed.
Show Background	If you turn on Show Background , the background color of the table is displayed.
Show Time	If you turn on Show Time , the time range of the query is displayed in the table.
Fixed Time	If you turn on Fixed Time , the time range of the query for the table is independent of the global time range of the dashboard.

• Parameters in the Standard Configurations section

Parameter	Description
Format	The display format of numeric values.
Unit	The unit of numeric values.
Number of Digits after Decimal Point	The decimal places of numeric values.
Display Name	The name of the table header. If you specify a value for Display Name, the value is used as the names for all headers in the table. If you want to change the name of a header, you must configure parameters on the Fields tab.
Color Scheme	 The color scheme of the table. The color scheme is applied to the background and text of the table. Valid values: Built-in: uses the built-in color scheme. Solid: uses the color that you select. Threshold: uses different colors for different values based on the specified thresholds for the values.

• Parameters in the Search & Analysis Settings section

Parameter	Description
Hide Field	The name of the field that is included in the results of a query statement and you want to hide in the table. For example, if you select time from the Hide Field drop-down list, the time field is hidden in the table.

• Parameters in the Table Configurations section

Parameter	Description
Display Mode	 The display mode of the table. Valid values: Paging: The table is displayed by page. You can configureltems Per Page to specify the number of rows that you want to display on each page. Contour: The table is displayed on one page. You can configureRow Height to specify the height of rows that you want to display on one page.
Display Header	If you turn on Display Header , the headers of the table are displayed.
Total	If you turn on Total , the total number of rows is displayed in the table.
Transparent Background	If you turn on Transparent Background , the background in the table is transparent.

• Parameters in the Column Settings section

Parameter	Description
Minimum Column Width	The minimum column width of the table. Unit: pixels. If you retain auto for Column Width, the column width of the table is greater than or equal to the value of Minimum Column Width.
Column Width	The column width of the table. Unit: pixels. By default, Log Service automatically calculates the column width of the table based on the size of the table and the value of Minimum Column Width. If you specify a value for Column Width, the value is used as the column width.
Max Value	The maximum value of the progress bar. If you retain auto for Max Value, the maximum value of the column is used. Max Value takes effect only when you setCell Display Mode to Progress, LCD Progress Bar, or Gradient Progress Bar.
Cell Display Mode	The display mode of cells.
Alignment Method	The alignment method of the content in cells.
Disable Sorting	If you turn on Disable Sorting , sorting is disabled.
Disable Searching	If you turn on Disable Search , searching is disabled.

Size	The font size of the content in cells.		
Parameters in the Threshold section			
Parameter	Description		
Threshold	The threshold of numeric values. If you set Color Scheme to Threshold and specify thresholds for values in the Threshold section, the values in the table are displayed in different colors based on the specified thresholds.		
Parameters in the Replace Variable section			
Parameter	Description		
Replace Variable	The settings of variable replacement. You can click AddReplace Variable to add a filter of the Replace Variable type to a single chart. After you configure the settings of variable replacement on the Common Settings tab, Log Service adds a filter in the upper-left corner of the chart. You can select a value from the filter drop-down list. After you select a value, Log Service adds a total value, Log Service automatically replaces the variable in the query statement of the chart with the variable value indicated by the value that you select, and performs a query and analysis operation. For more information, see Example 2: Configure variable replacement.		
Parameters in the Mapping Value section			
Parameter	Description		
Mapping Value	The text or icon that you want to use to replace a specified value in the table. For example, if you set Value to 200 , Mapping Type to Text , and Mapping Value to Success , all values of 200 in the table are replaced with Success .		
Parameters in the Documentation section			

Parameter	Description
Add Documentation Link	The button that allows you to specify custom document links and descriptions. After you configure the settings, the specified information is displayed in the upper right corner of the table.

Configurations on the Fields tab

You can configure personalized display settings for the results of a single query statement or for a single column of data in the results. For more information about the parameters on the Fields tab, see Configurations on the Common Settings tab.

For example, in the **A** > **pv** section, you can configure settings for the **pv** field in the results of Query Statement A. The values of the **pv** field are highlighted after you select Text Highlight from the Column Settings > Cell Display Mode drop-down list and are displayed in different colors based on the specified thresholds.

			Display Raw D	Ata 5 Minutes(Relative)	pply	Common Settings Fields Interaction Event	5
test 5 Minutes(Relative)					:	~ A > pv (3)	1
time \$ Q	pv \$ Q	icon ‡ Q	database $\ \ $	hide	Q	Threshold	Û
1656401940	11397	compute	database	hide			
1656402000	106611	compute	database	hide		● 12000 Greater Than ∨	Delete
1656402060	116423	compute	database	hide		Add Threshold	
1656402120	114191	compute	database	hide		Standard Configurations > Color Scheme	Ē
1656402180	75714	compute	database	hide		Threshold	\sim
1656402240	81511	compute	database	hide		Column Settings > Cell Display Mode	Ē
						Text Highlight	\sim

Drill-down events

You can configure a drill-down event for the results of a single query statement or for a single column of data in the results to analyze data from a finergrained dimension. Drill-down events include events to open a Logstore, open a saved search, open a dashboard, open trace analysis, open trace details, and customize an HTTP link. For more information, see Drill-down events.

For example, in the **A** > **pv** section, you can configure an Open Logstore drill-down event for the **pv** field in the results of Query Statement A. After you configure the event, you can click a value in the **pv** column of the table and click **Open Logstore**. Then, you are navigated to the Logstore that you specify.

			Display Raw D	Apply 5 Minutes(Relative)	Common Settings Fields	Interaction Events
test 5 Minutes(Relative)				:	∧ A > pv (1)	Ê
time \$ Q	pv \$ Q	icon ‡ Q	database $ au$ Q	hide 🌣 Q	Open Logstore	之 首
1656401940	11397	comnute	database	hide	Add Event	
1656402000	Custom Event	ute	database	hide	Add Field	
1656402060	110960	compute	database	hide		
1656402120	<u>114191</u>	compute	database	hide		
1656402180	75714	compute	database	hide		
1656402240	<u>81511</u>	compute	database	hide		

4.4.15.2.7. Line chart (Pro)

You can use a line chart (Pro) to visualize the results of multiple query statements. You can also configure personalized display settings on the Fields tab. This topic describes the basic configurations of a line chart (Pro).

Introduction

A line chart is used to analyze the value changes of a categorical variable over a continuous time range. In most cases, the analysis is based on a

specified time range. You can intuitively view value trends. You can use a line chart to analyze the following change characteristics of values over a specified time range:

- Increment or decrement
- Increment or decrement rate
- Increment or decrement pattern, such as periodicity
- Peak value and bottom value

Line charts are the optimal choice for you to analyze value changes over a time range. You can also use a line chart to analyze the value changes of multiple fields in multiple lines over the same time range. Then, you can analyze the relationships between the fields. For example, the values of the fields are directly or inversely proportional to each other.

For more information about how to create a line chart, see Add a chart (Pro) to a dashboard.

Configurations on the Common Settings tab

You can configure global settings for a line chart on the Common Settings tab.

• Parameters in the Basic Configurations section

Parameter	Description
Title	The title of the line chart.
Show Title	If you turn on Show Title , the title of the line chart is displayed.
Show Border	If you turn on Show Border , the borders of the line chart are displayed.
Show Background	If you turn on Show Background, the background color of the line chart is displayed.
Show Time	If you turn on Show Time , the query time range of the line chart is displayed.
Fixed Time	If you turn on Fixed Time , the query time range of the line chart is independent of the global time range of the dashboard.

• Parameters in the Standard Configurations section

Parameter	Description
Format	The display format of numeric values.
Unit	The unit of numeric values.
Number of Digits after Decimal Point	The decimal places of numeric values.
Display Name	The name of the legend. If you specify a value for Display Name, the value is used as the name for all items in the legend that is displayed in the line chart. If you want to change the name of an item in the legend, you must configure parameters on the Fields tab.
Color Scheme	The color scheme of the line chart. The color scheme is applied to the background and legend of the line chart. Valid values: • Built-in: uses the built-in color scheme. • Solid: uses the color that you select.

• Parameters in the Configure Query and Analysis section

Parameter	Description
Axis X Field	Select a field as the x-axis. By default, Log Service automatically selects an appropriate field as the x-axis.
Axis Y Field	Select a field as the y-axis. By default, Log Service automatically selects an appropriate field as the y-axis.

Parameters in the Data Configuration section

Parameter	Description
Data Completion	If you turn on Data Completion , Log Service automatically replaces missing values with a specified value from the first data record within the time window specified by Completion window.
Completion window	The time window of data for which Log Service automatically replaces missing values with a specified value. Minimum value: 0. Unit: seconds. By default, Log Service automatically specifies the value of Completion window.
Text-substituted Value	The value that you want to use to replace missing values. Default value: 0, which indicates that Log Service replaces missing values with 0.

• Parameters in the Legend Configurations section

Parameter	Description
Display Legend	If you turn on Display Legend , the legend of the line chart is displayed.
Legend	The position of the legend in the line chart.

٠

Actions	 The data display effect when you click a legend item. Valid values: Single: If you click a legend item, only the data corresponding to the legend item is displayed in the line chart. Switch: If you click a legend item, the data corresponding to the legend item is hidden or displayed in the line chart.
Maximum Width (Height)%	The maximum width and height of the legend.
Parameters in the Tooltip Configurations section	

Parameter	Description
Sorting Order	The sorting method of data.
Parameters in the Axis X section	
Parameter	Description
Display Axis X	If you turn on Display Axis X , the x-axis of the line chart is displayed.
Axis X Title	The title of the x-axis.

By default, Log Service automatically specifies the height of the x-axis.

The height of the x-axis.

• Parameters in the Axis Y section

Axis X Height

Parameter	Description
Display Axis Y	If you turn on Display Axis Y , the y-axis of the line chart is displayed.
Axis Y Title	The title of the y-axis.
Axis Y Position	The position of the y-axis.
Axis Y Width	The width of the y-axis. By default, Log Service automatically specifies the width of the y-axis.
Max Value	The maximum value of the y-axis. By default, Log Service automatically specifies the maximum value of the y-axis.
Minimum	The minimum value of the y-axis. By default, Log Service automatically specifies the minimum value of the y-axis.
Elastic Maximum Value	The elastic maximum value of the y-axis. The elastic maximum value takes effect only when all values of the y-axis are less than the elastic maximum value. By default, Log Service automatically specifies the elastic maximum value of the y-axis.
Elastic Minimum Value	The elastic minimum value of the y-axis. The elastic minimum value takes effect only when all values of the y-axis are greater than the elastic minimum value. By default, Log Service automatically specifies the elastic minimum value of the y-axis.
Axis Y ID	The ID of the y-axis. This parameter does not take effect if you configure only one y-axis on th €Common Settings tab. If you want to configure multiple y-axes, you must configure parameters on th €rields tab. The ID of the y-axis is a string. A y-axis that uses a specified ID represents a unique y-axis in a line chart. The configuration of the ID for the y-axis takes precedence over the unit that you specify in th §tandard Configurations section. For example, if two y-axes use the same unit, the two y-axes are merged into one y- axis. If two y-axes use the same unit but different IDs, they are displayed as two y-axes.

• Parameters in the Chart Configurations section

Parameter	Description
Chart Type	The type of the line chart.
Connection Mode	The connection mode of the line in the line chart.
Connector Width	The width of the line.
Show Points	Specifies whether to display points in the line chart.
Transparency	The transparency of the line chart.
Point Size	The size of each point in the line chart.
Gradient Mode	 The gradient mode of the line chart. Valid values: Transparency: If you select this mode, the line chart uses a color gradient that is affected by the line color and the value of Transparency. No Gradient: If you select this mode, the line chart does not use a color gradient. The line color is used in the line chart.
Parameters in the Replace Variable section	

Parameter

Description

Replace Variable	The settings of variable replacement. You can click AddReplace Variable to add a filter of the Replace Variable type to a single chart. After you configure the settings of variable replacement on the Common Settings tab, Log Service adds a filter in the upper-left corner of the chart. You can select a value from the filter drop-down list. After you select a value, Log Service automatically replaces the variable in the query statement of the chart with the variable value indicated by the value that you select, and performs a query and analysis operation. For more information, see Example 2: Configure variable replacement.
Parameters in the Documentation section	
Parameter	Description
Add Documentation Link	The button that allows you to specify custom document links and descriptions. After you configure the

settings, the specified information is displayed in the upper right corner of the line chart.

Configurations on the Fields tab

You can configure personalized display settings for the results of a single query statement or for a single column of data in the results. For more information about the parameters on the Fields tab, see Common Settings tab.

For example, in the A > pv section, you can configure settings for the pv field in the results of Query Statement A. In the B section, you can configure settings for the results of Query Statement B. In the following figure, the green line indicates the pv field in the results of Query Statement A, and the blue line indicates the results of Query Statement B. You can set Connector Width to 6 for the green line and 2 for the blue line.

Drill-down events

Drill-down events are used to analyze a single field or the results of a single query statement from a finer-grained dimension. Drill-down events include events to open a Logstore, open a saved search, open a dashboard, open trace analysis, open trace details, and customize an HTTP link. For more information, see Drill-down events.

For example, in the **A** > **pv** section, you can configure an Open Logstore drill-down event for the **pv** field in the results of Query Statement A. After you configure the event, you can click a point on the green line of the line chart and click **Open Logstore**. Then, you are navigated to the Logstore that you specify.

4.4.15.2.8. Flow chart (Pro)

You can use a flow chart (Pro) to visualize the results of multiple query statements. You can also configure personalized display settings on the Fields tab. This topic describes the basic configurations of a flow chart (Pro).

Introduction

A flow chart, which is known as a ThemeRiver, is a stacked area chart around a central axis. Lines of different colors in a flow chart represent different categories. By default, the time information of data is mapped to the x-axis of a flow chart. This enables the three-dimensional visualization of data. You can switch from a flow chart to a column chart below Chart Types.

For more information about how to create a flow chart, see Add a chart (Pro) to a dashboard.

Configurations on the Common Settings tab

• Parameters in the Basic Configurations section

Parameter	Description
Title	The title of the flow chart.
Show Title	If you turn on Show Title , the title of the flow chart is displayed.
Show Border	If you turn on Show Border , the borders of the flow chart are displayed.
Show Background	If you turn on Show Background , the background color of the flow chart is displayed.
Show Time	If you turn on Show Time , the time range of a query is displayed in the flow chart.
Fixed Time	If you turn on Fixed Time , the time range of a query for the flow chart is independent of the global time range of the dashboard.

· Parameters in the Standard Configurations section

Parameter	Description
Format	The display format of numeric values.
Unit	The unit of numeric values.
Number of Digits after Decimal Point	The decimal places of numeric values.
Display Name	The name of the legend. If you specify a value for Display Name, the value is used as the names of all legends in the flow chart. If you want to change the name of a legend, you must configure parameters on the Fields tab.
Color Scheme	The color scheme of the flow chart. The color scheme is applied to the background and legends of the flow chart. • Built-in: uses the built-in color scheme. • Solid: uses the color that you select.

• Parameters in the Configure Query and Analysis section

Parameter	Description
Axis X Field	Select a field as the x-axis. By default, Log Service automatically selects an appropriate field as the x-axis.

Axis Y Field	Select a field as the y-axis. By default, Log Service automatically selects an appropriate field as the y-axis.
Aggregate Column	Select a field based on which you want to further categorize and aggregate data.

• Parameters in the Data Configuration section

Parameter	Description
Data Completion	If you turn on Data Completion , Log Service automatically replaces missing values with a specified value from the first data record within the time window specified by Completion window.
Completion window	The time window of data for which Log Service automatically replaces missing values with a specified value. Minimum value: 0. Unit: seconds. By default, Log Service automatically specifies the value of Completion window.
Text-substituted Value	The value that you want to use to replace missing values. Default value: 0, which indicates that Log Service replaces missing values with 0.

Parameters in the Tooltip Configurations section

Parameter	Description
Sorting Order	The sorting method of data.

Parameters in the Legend Configurations section

Parameter	Description
Display Legend	If you turn on Display Legend , the legends of the flow chart are displayed.
Legend	The position of the legend.
Actions	 The data display effect when you click a legend. Single: When you click a legend, only the data corresponding to the legend is displayed in the flow chart. Switch: When you click a legend, the data corresponding to the legend is hidden or displayed in the flow chart.
Maximum Width (Height)%	The maximum width and height of the legend.

• Parameters in the Axis X section

Parameter	Description
Display Axis X	If you turn on $\ensuremath{\textbf{Display}}\xspace{\ensuremath{\textbf{Axis}}\xspace{\ensuremath{\textbf{X}}}\xspace, \ensuremath{\textbf{the}}\xspace{\ensuremath{\textbf{S}}\xspace{\ensuremath{\textbf{Display}}\xspace{\ensuremath{\textbf{Axis}}\xspace{\ensuremath{\textbf{X}}}\xspace{\ensuremath{\textbf{S}}\xspace{\ensuremath{\textbf{S}}\xspace{\ensuremath{\textbf{S}}\xspace{\ensuremath{\textbf{S}}\xspace{\ensuremath{\textbf{S}}\xspace{\ensuremath{\textbf{S}}\xspace{\ensuremath{\textbf{S}}\xspace{\ensuremath{\textbf{D}}\xspace{\ensuremath{\textbf{S}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\textbf{S}}\xspace{\ensuremath{\textbf{S}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\xspace{\ensuremath{\s}}\$
Axis X Title	The title of the x-axis.
Axis X Height	The height of the x-axis. By default, Log Service automatically specifies the height of the x-axis

• Parameters in the Axis Y section

Parameter	Description
Display Axis Y	If you turn on Display Axis Y , the y-axis of the flow chart is displayed.
Axis Y Title	The title of the y-axis.
Axis Y Position	The position of the y-axis.
Axis Y Width	The width of the y-axis. By default, Log Service automatically specifies the width of the y-axis.
Max Value	The maximum value of the y-axis. By default, Log Service automatically specifies the maximum value of the y-axis.
Minimum	The minimum value of the y-axis. By default, Log Service automatically specifies the minimum value of the y-axis.
Elastic Maximum Value	The elastic maximum value of the y-axis. The elastic maximum value takes effect only when all values of the y-axis are less than the elastic maximum value. By default, Log Service automatically specifies the elastic maximum value of the y-axis.
Elastic Minimum Value	The elastic minimum value of the y-axis. The elastic minimum value takes effect only when all values of the y-axis are greater than the elastic minimum value. By default, Log Service automatically specifies the elastic minimum value of the y-axis.
Axis Y ID	The ID of the y-axis. In most cases, this parameter does not take effect if you configure only one y-axis in Common Settings . If you want to configure multiple y-axes, you must configure parameters on th Fields tab. The ID of the y-axis is a string. A y-axis that has a specified ID represents a unique y-axis in a flow chart. The configuration of the ID for the y-axis takes precedence over the unit that you specify in Standard Configurations . For example, if two y-axes have the same unit, the two y-axes are merged into one y-axis. If two y-axes have the same unit, the two y-axes.
• Parameters in the Chart Configurations section

Parameter	Description
Chart Type	The type of the flow chart. Once The available parameters in the Chart Configurations section vary based on the type of the flow chart that you specify. The following parameters are available if you specify Line for Chart Type.
Connection Mode	The connection mode of the line in the flow chart.
Connector Width	The width of the line.
Show Points	Specifies whether to display points in the chart.
Transparency	The transparency of the flow chart.
Point Size	The size of the point in the flow chart.
Gradient Mode	 The gradient mode of the flow chart. Transparency: If you select this mode, the flow chart has a color gradient that is affected by the line color and the value of Transparency. No Gradient: If you select this mode, the flow chart does not have a color gradient. The line color is used in the flow chart.

Parameters in the Replace Variable section

Parameter	Description
Replace Variable	The settings of variable replacement. You can click AddReplace Variable to add a filter of the Replace Variable type to a single chart. After you configure the settings of variable replacement on the Common Settings tab, Log Service adds a filter in the upper-left corner of the chart. You can select a value from the filter drop- down list. After you select a value, Log Service automatically replaces the variable in the query statement of the chart with the variable value indicated by the value that you select, and performs a query and analysis operation. For more information, see Example 2: Configure variable replacement.

Parameters in the Documentation section

Parameter	Description
Add Documentation Link	The button that allows you to specify custom document links and descriptions. After you configure the settings, the specified information is displayed in the upper right corner of the flow chart.

Configurations on the Fields tab

You can configure personalized display settings for the results of a single query statement or for a single column of data in the results. For more information about the parameters on the Fields tab, see Common Settings tab.

For example, in the **A** section, you can configure personalized display settings for the results of Query Statement A. In the **B** section, you can configure personalized display settings for the results of Query Statement B. You can specify the thickness of each line to distinguish the results of the two query statements. In the following figure, the thick line indicates the results of Query Statement A, and the thin line indicates the results of Query Statement B.

Drill-down events

Drill-down events are used to analyze a single field or the results of a single query statement from a finer-grained dimension. Drill-down events include events to open a Logstore, open a saved search, open a dashboard, open trace analysis, open trace details, and customize an HTTP link. For more information, see Drill-down events.

For example, in the **A** section, you can configure an **Open Logstore** drill-down event for the results of Query Statement A. After you configure the event, you can click a point on a line of the flow chart and click **Open Logstore**. Then, you are navigated to the Logstore that you specify.

4.4.15.2.9. Column chart (Pro)

A column chart (Pro) can be a column chart or a bar chart. You can use a column chart (Pro) to visualize the results of multiple query statements. You can also configure personalized display settings on the Fields tab. This topic describes the basic configurations of a column chart (Pro).

Introduction

A column chart uses vertical bars to show the values of different categories. You can use a column chart to display the number of values in each category. A bar chart uses horizontal bars. Bar charts are often used in top-N analysis.

For more information about how to create a column chart or a bar chart, see Add a chart (Pro) to a dashboard.

Configurations on the Common Settings tab

You can configure global settings for a column chart on the Common Settings tab.

Parameters in the Basic Configurations section

Parameter	Description
Title	The title of the column chart.
Show Title	If you turn on Show Title , the title of the column chart is displayed.
Show Border	If you turn on Show Border , the borders of the column chart are displayed.
Show Background	If you turn on Show Background, the background color of the column chart is displayed.
Show Time	If you turn on Show Time, the query time range of the column chart is displayed.
Fixed Time	If you turn on Fixed Time , the query time range of the column chart is independent of the global time range of the dashboard.

· Parameters in the Standard Configurations section

Parameter	Description
Format	The display format of numeric values.
Unit	The unit of numeric values.
Number of Digits after Decimal Point	The decimal places of numeric values.
Display Name	The name of the legend. If you specify a value for Display Name, the value is used as the names of all legends in the column chart. If you want to change the name of a legend, you must configure parameters on the Fields tab.
Color Scheme	The color scheme of the column chart. The color scheme is applied to the background and legends of the column chart. Valid values: Built-in: uses the built-in color scheme. Solid: uses the color that you select.

• Parameters in the Column Settings section

Parameter	Description
Direction	The direction of the columns in the chart. The direction is used to distinguish between column charts and bar charts. Valid values: • Vertical: indicates a column chart. • Horizontal: indicates a bar chart.
Group Width	The width of the group.
Column Width	The width of the column.
Rotation Angle	The direction of the text in the x-axis. If you set Direction to Vertical , you must configure this parameter.
Show Value	Specifies whether to display numeric values in the column chart.
Value Font Size	The size of the text displayed in the column chart.
Connector Width	The width of the border.
Transparency	The transparency of the column chart.
Gradient Mode	 The gradient mode of the column chart. Valid values: Transparency: If you select this mode, the column chart has a color gradient that is affected by the column color and the value of Transparency. No: If you select this mode, the column chart does not have a color gradient. The column color is used in the column chart.

Parameters in the Configure Query and Analysis section

Parameter	Description
Axis X Field	Select a field as the x-axis. By default, Log Service automatically selects an appropriate field as the x-axis.
Axis Y Field	Select a field as the y-axis. You can specify multiple fields to generate a grouped column chart. By default, Log Service automatically selects an appropriate field as the y-axis.

• Parameters in the Legend Configurations section

Parameter	Description
Display Legend	If you turn on Display Legend , the legends of the column chart are displayed.
Legend	The position of the legend.
Actions	 The data display effect when you click a legend item. Valid values: Single: If you click a legend item, only the data corresponding to the legend item is displayed in the column chart. Switch: If you click a legend item, the data corresponding to the legend item is hidden or displayed in the column chart.
Maximum Width (Height)%	The maximum width and height of the legend.

Parameters in the Tooltip Configurations section

Parameter	Description
Sorting Order	The sorting method of data. When you move the pointer over a column whose data you want to view, the data is displayed based on the sorting method that you specify.
Display Mode	The display mode of data. When you move the pointer over a column whose data you want to view, the data is displayed based on the display mode that you specify.

• Parameters in the Axis X section

Parameter	Description
Display Axis X	If you turn on Display Axis X , the x-axis of the column chart is displayed.
Axis X Title	The title of the x-axis.
Axis X Height	The height of the x-axis. By default, Log Service automatically specifies the height of the x-axis.

• Parameters in the Axis Y section

Parameter	Description
Display Axis Y	If you turn on Display Axis Y , the y-axis of the column chart is displayed.
Axis Y Title	The title of the y-axis.
Axis Y Position	The position of the y-axis.
Axis Y Width	The width of the y-axis. By default, Log Service automatically specifies the width of the y-axis.
Max Value	The maximum value of the y-axis. By default, Log Service automatically specifies the maximum value of the y-axis.
Minimum	The minimum value of the y-axis. By default, Log Service automatically specifies the minimum value of the y-axis.
Elastic Maximum Value	The elastic maximum value of the y-axis. The elastic maximum value takes effect only when all values of the y-axis are less than the elastic maximum value. By default, Log Service automatically specifies the elastic maximum value of the y-axis.
Elastic Minimum Value	The elastic minimum value of the y-axis. The elastic minimum value takes effect only when all values of the y-axis are greater than the elastic minimum value. By default, Log Service automatically specifies the elastic minimum value of the y-axis.
Axis Y ID	The ID of the y-axis. In most cases, this parameter does not take effect if you configure only one y-axis in Common Settings . If you want to configure multiple y-axes, you must configure parameters on th Fields tab. The ID of the y-axis is a string. A y-axis that has a specified ID represents a unique y-axis in a column chart. The configuration of the ID for the y-axis takes precedence over the unit that you specify in Standard Configurations . For example, if two y-axes have the same unit, the two y-axes are merged into one y-axis. If two y-axes have the same unit but different IDs, they are displayed as two y-axes.
Parameters in the Benlace Variable section	

and the replace values section	
Parameter	Description
Replace Variable	The settings of variable replacement. You can click AddReplace Variable to add a filter of the Replace Variable type to a single chart. After you configure the settings of variable replacement on the Common Settings tab, Log Service adds a filter in the upper-left corner of the chart. You can select a value from the filter drop-down list. After you select a value, Log Service adds a fuller in the upper-left corner of the chart. You can select a value from the filter drop-down list. After you select a value, Log Service automatically replaces the variable in the query statement of the chart with the variable value indicated by the value that you select, and performs a query and analysis operation. For more information, see Example 2: Configure variable replacement.

• Parameters in the Documentation section

Parameter	Description
Add Documentation Link	The button that allows you to specify custom document links and descriptions. After you configure the settings, the specified information is displayed in the upper right corner of the column chart.

Configurations on the Fields tab

You can configure personalized display settings for the results of a single query statement or for a single column of data in the results. For more information about the parameters on the Fields tab, see Common Settings tab.

For example, in the A > uv section, you can configure settings for the uv field in the results of Query Statement A. You can add a different y-axis so that the uv and avg fields are represented by different y-axes in the same column chart. In the following figure, the left y-axis displays the values of the avg field, and the right y-axis displays the values of the uv field.

Drill-down events

Drill-down events are used to analyze a single field or the results of a single query statement from a finer-grained dimension. Drill-down events include events to open a Logstore, open a saved search, open a dashboard, open trace analysis, open trace details, and customize an HTTP link. For more information, see Drill-down events.

For example, in the **A** section, you can configure an **Open Logstore** drill-down event for the results of Query Statement A. After you configure the event, you can click a point in the column chart and click **Open Logstore**. Then, you are navigated to the Logstore that you specify.

4.4.15.2.10. Single value chart (Pro)

You can use a single value chart (Pro) to visualize the results of multiple query statements. You can also configure personalized display settings on the Fields tab. This topic describes the basic configurations of a single value chart (Pro).

Introduction

A single value chart (Pro) contains one or more single value charts. Each single value chart is used to display a single numeric value. The following figure shows sample single value charts. Each colored rectangle represents a single value chart. For more information about how to create a single value chart ,see Add a chart (Pro) to a dashboard.

Configurations on the Common Settings tab

On the Common Settings tab, you can configure global settings for a single value chart (Pro).

Parameters in the Basic Configurations section

Parameter	Description
Title	The title of the single value chart (Pro).
Show Title	If you turn on Show Title , the title of the single value chart (Pro) is displayed.
Show Border	If you turn on Show Border , the borders of the single value chart (Pro) are displayed.
Show Background	If you turn on Show Background, the background color of the single value chart (Pro) is displayed.
Show Time	If you turn on Show Time , the query time range of the single value chart (Pro) is displayed.
Fixed Time	If you turn on Fixed Time , the query time range of the single value chart (Pro) is independent of the global time range of the dashboard.

• Parameters in the Standard Configurations section

Parameter	Description
Format	The display format of numeric values.
Unit	The unit of numeric values.
Number of Digits after Decimal Point	The decimal places of numeric values.
Display Name	The display name of a single value chart. If you specify a value for Display Name, the value is used as the name of each single value chart in the single value chart (Pro). If you want to change the name of a single value chart, you must configure parameters on the Fields tab.
Color Scheme	 The color scheme that specifies the background color of the single value chart (Pro). Valid values: Built-in: uses the built-in color scheme. Solid: uses the color that you select. Threshold: uses different colors for different values based on the specified thresholds for the values.

• Parameters in the Data Configuration section

Parameter	Description
Display Mode	 The display mode of a single value chart. Valid values: Calculation Result: displays the calculation result of all values of a field based on the query results. All Values: displays all values of a field based on the query results.
Function	If you set the Display Mode parameter to Calculation Result , you must select a function to calculate query results. For example, if you set the Function parameter to Maximum , the maximum value of a field is displayed based on the query results.
Limit	If you set the Display Mode parameter to All Values , you can configure the Limit parameter to specify the number of single value charts that you want to display in the single value chart (Pro).
Layout Mode	 The layout of single value charts. Valid values: Automatic: The single value charts are arranged in an adaptive manner. Horizontal: The single value charts are arranged from left to right. You can specify the minimum width for single value charts. Vertical: The single value charts are arranged from top to bottom. You can specify the minimum height for single value charts.

• Parameters in the Chart Style section

Parameter	Description
Text Mode	 The text that you want to display on a single value chart. Valid values: Auto: displays text based on the query results. Value: displays only a numeric value. Value and Title: displays a numeric value and a title. Title: displays only a title. None: No content is displayed.
Color Mode	 The color of a single value chart. Valid values: None: displays text in black. Value: displays only the mini-charts and numeric values of the single value chart in the specified color. Background: displays the background and mini-charts of the single value chart in the specified color.

Image Mode	Specifies whether to display mini-charts. Valid values: None: displays only numeric values. Mini-chart: displays mini-charts. Once The Mini-chart parameter takes effect only when the Display Mode parameter is set to Calculation Result.
Text Alignment	 The mode in which the text of a single value chart is aligned. Valid values: Automatic: Text is aligned in an adaptive manner. Center: Text is located in the center of a single value chart.

Parameters in the Search & Analysis Settings section

Parameter	Description
Displayed Field	The field whose value you want to display on a single value chart.
Compared Field	The field whose value you want to compare with the value of the displayed field.
Compare Value Format	The format of the value of the compared field.
Comparison Result Description	The description of the value of the compared field.
Number of Digits after Decimal Point	The decimal places of numeric values.
Description Font Size	The font size of the description.

· Parameters in the Threshold section

Parameter	Description
Threshold	The thresholds of numeric values. If you set the Color Scheme parameter to Threshold and specify thresholds in the Threshold section, the text on a single value chart (Pro) is displayed in different colors based on the specified thresholds.

Parameters in the Mapping Value section

Parameter	Description
Mapping Value	The text or icon that is used to replace a specified value in a single value chart (Pro). For example, if you set the Value parameter to 200 , the Mapping Type parameter to Text , and the Mapping Value parameter to Success , all values of 200 on single value charts are replaced by Success .

• Parameters in the Replace Variable section

Parameter	Description
Replace Variable	The settings of variable replacement. You can click AddReplace Variable to add a filter of the Replace Variable type to a single value chart (Pro). After you configure the settings of variable replacement on the Common Settings tab, Log Service adds a filter in the upper-left corner of the chart. You can select a value from the filter drop-down list. After you select a value, Log Service automatically replaces the variable in the query statement of the chart with the variable value indicated by the value that you select, and performs a query and analysis operation. For more information, see Example 2: Configure variable replacement.

• Parameters in the Documentation section

Parameter	Description
Add Documentation Link	The button that allows you to specify custom document links and descriptions. After you configure the settings, the specified information is displayed in the upper-right corner of the single value chart (Pro).

Configurations on the Fields tab

You can configure personalized display settings for the results of a single query statement or for a single column of data in the results. For more information about the parameters on the Fields tab, see Common Settings tab.

In this example, five query statements named A, B, C, D, and E are added to calculate the number of requests for each request method. The results of the query statements are displayed on a single value chart (Pro). You can click the Single Value Chart Pro icon, choose **Common Settings > Search & Analysis Settings**, and then specify a displayed field for each query statement to generate five single value charts. On the **Fields** tab, configure different colors for the single value charts. The following figure shows the single value charts and related settings.

- A > GET indicates that the GET column in the result of the query statement A is selected as the displayed field. The color of the single value chart is purple.
- B > POST indicates that the POST column in the result of the query statement B is selected as the displayed field. The color of the single value chart is blue.
- C > PUT indicates that the PUT column in the result of the query statement C is selected as the displayed field. The color of the single value chart is orange.
- D > D indicates that the D column in the result of the query statement D is selected as the displayed field. The color of the single value chart is yellow. DELETE is a keyword in SQL syntax and cannot be used as a column name in analytic statements. In this case, you can specify DELETE as the display name for the D column on the Fields tab.
- E > HEAD indicates that the HEAD column in the result of the query statement E is selected as the displayed field. The color of the single value chart is red.

4.4.15.2.11. Pie chart (Pro)

This topic describes the basic configurations of a pie chart (Pro).

Introduction

A pie chart is used to show the percentage of each category of data. The arc length of each slice in a pie chart is proportionate to the quantity that is represented by each category. A pie chart is divided into multiple slices based on the percentages of categories. Each slice shows the percentage of a category of data. The sum of all percentages is equal to 100%.

Log Service provides the following types of standard pie charts: pie chart, donut chart, and polar area chart. The following list describes the donut and polar area charts:

• Donut chart

- A donut chart is a variant of a pie chart and has a hollow center. Compared with a pie chart, a donut chart provides the following advantages:
- Displays more information, such as the total number of occurrences of all field values.
- Allows you to compare data between two donut charts based on ring lengths.
- Polar area chart

A polar area chart is a column chart in the polar coordinate system. Each category of data is represented by a slice with the same angle, and the radius of each slice varies based on the value. Compared with a pie chart, a polar area chart provides the following advantages:

- If the result that a query statement returns can be classified into no more than 10 categories, you can use a pie chart to display the query result. If the result can be classified into 10 to 30 categories, you can use a polar area chart to display the query result.
- A polar area chart enlarges the differences among the values of categories because the area of a slice correlates with the square of the radius. The polar area chart is suitable for the comparison of values that have small differences with one another.
- A circle can be used to display periodic data. You can use a polar area chart to analyze value changes in different periods, such as weeks and months.

For more information about how to create a pie chart, see Add a chart (Pro) to a dashboard.

Configurations on the Common Settings tab

You can configure global settings for a pie chart on the Common Settings tab.

• Parameters in the Basic Configurations section

Parameter	Description
Title	The title of the pie chart.
Show Title	If you turn on Show Title , the title of the pie chart is displayed.
Show Border	If you turn on Show Border , the borders of the pie chart are displayed.
Show Background	If you turn on Show Background, the background color of the pie chart is displayed.
Show Time	If you turn on Show Time , the query time range of the pie chart is displayed.
Fixed Time	If you turn on Fixed Time , the query time range of the pie chart is independent of the global time range of the dashboard.

• Parameters in the Standard Configurations section

Parameter	Description		
Format	The display format of numeric values.		
Unit	The unit of numeric values.		
Number of Digits after Decimal Point	The decimal places of numeric values.		
Display Name	The display name.		
Color Scheme	 The color scheme of the pie chart. Valid values: Built-in: uses the built-in color scheme. Solid: uses the color that you select. 		

• Parameters in the Configure Query and Analysis section

Parameter	Description	
Category	The field that is used to categorize data.	
Value Column	The field that specifies the numeric values to be displayed.	

• Parameters in the Pie Chart Configurations section

Parameter	Description	
Pie Chart Type	The type of the pie chart. Valid values: Pie Chart, Donut Chart, and Polar Area Chart.	
Tick Text Format	 The format of the scale text that is displayed for the slices in the pie chart. Valid values: Percentage: displays the scale text in the Percentage format. Example: 1.98%. Category: Percentage: displays the scale text in the Category: Percentage format. Example: PUT: 1.98%. Category:Numeric Value (Percentage): displays the scale text in the Category:Numeric Value (Percentage): displays the scale text in the Category:Numeric Value (Percentage). 	
Show Scale Text	If you turn on Show Scale Text , the scale text of each slice in the pie chart is displayed.	

Parameters in the Legend Configurations section

Parameter	Description	
Display Legend	If you turn on Display Legend , the legends of the pie chart are displayed.	
Legend	The position of the legend.	
Actions	 The data display effect when you click a legend item. Valid values: Single: If you click a legend item, only the data corresponding to the legend item is displayed in the pie chart. Switch: If you click a legend item, the data corresponding to the legend item is hidden or displayed in the pie chart. 	
Maximum Width (Height)%	The maximum width and height of the legend.	
Parameters in the Replace Variable section		

Parameter	Description	
Replace Variable	The settings of variable replacement. You can click AddReplace Variable to add a filter of the Replace Variable type to a single chart. After you configure the settings of variable replacement on the Common Settings tab, Log Service adds a filter in the upper-left corner of the chart. You can select a value from the filter drop-down list. After you select a value, Log Service automatically replaces the variable in the query statement of the chart with the variable value indicated by the value that you select, and performs a query and analysis operation. For more information, see Example 2: Configure variable replacement.	

• Parameters in the Documentation section

Parameter	Description
Add Documentation Link	The button that allows you to specify custom document links and descriptions. After you configure the settings, the specified information is displayed in the upper-right corner of the pie chart.

Drill-down events

Drill-down events are used to analyze a single field or the results of a single query statement from a finer-grained dimension. Drill-down events include events to open a Logstore, open a saved search, open a dashboard, open trace analysis, open trace details, and customize an HTTP link. For more information, see Drill-down events.

For example, you can configure an **Open Logstore** drill-down event for the result of Query Statement **A**. After you configure the event, click a point on the pie chart (Pro) and click **Open Logstore**. Then, you are redirected to the Logstore that you specified in the drill-down event.

4.4.15.3. Dashboard

4.4.15.3.1. Overview

A dashboard provided by Log Service is a platform where you can analyze data in real time. You can add multiple charts to a dashboard for data analysis. Each chart is a visualized search and analytic statement.

A dashboard allows you to view the charts of multiple search and analytic statements at one time. When you open or refresh the dashboard, the statements of the charts run automatically.

After you add a chart to a dashboard, you can configure Configure a drill-down event for the chart. Then you can click the chart on the dashboard to further analyze data and obtain more fine-grained analysis results.

Limits

- You can create a maximum of 50 dashboards for a project.
- Each dashboard can contain a maximum of 50 analysis charts.

Features

- A dashboard has two modes: display mode and edit mode.
- Manage a dashboard in display mode
- In the display mode, you can configure multiple display settings on the dashboard page.
- Dashboard: You can specify the time range, the automatic refresh interval, full screen, and the display mode of the title for the dashboard, configure alerts for all charts on the dashboard, and filter chart data based on the Add a filter.
- Chart: You can view the analysis details of a specified chart, specify the time range and configure alerts for the chart, download logs and the chart, and check whether drill-down analysis is configured for the chart.
- Manage a dashboard in edit mode

In the edit mode, you can change the configurations of the dashboard and charts.

- Dashboard: You can use a dashboard as a canvas and add Manage a Markdown chart, custom charts, text, icons, and other chart elements to the dashboard. You can also add lines between chart elements that are self-adaptive to the positions of the charts. You can also add Add a filter, which can be used to filter chart data in the display mode. In addition, you can configure display gridlines to help arrange chart elements such as icons in an orderly manner.
- Chart: You can also edit a chart on the dashboard. You can modify the statement, properties, and interactive behavior such as drill-down analysis of the chart.

4.4.15.3.2. Create and delete a dashboard

This topic describes how to create a dashboard. After you create a dashboard, you can follow, view, and delete the dashboard.

Prerequisites

The indexing feature is enabled and indexes are configured. For more information, see Configure indexes.

Create a dashboard

1. Log on to the Log Service console

- 2. In the Projects section, click the name of the project in which you want to create a dashboard.
- 3. In the left-side navigation pane, click the icon.

4. Click the plus sign (+) to create a dashboard.

5. In the Add to New Dashboard dialog box, choose Layout Mode, enter a name for the dashboard in the Dashboard Name field and click OK.

Related operations

After you create a dashboard, you can follow, view, and delete the dashboard.

In the Dashboard list, find the dashboard that you created and choose
 Delete to delete the dashboard.

() Important

After you delete a dashboard, the dashboard cannot be restored. Proceed with caution.

- In the Dashboard list, find the dashboard that you created and choose > Details to view the dashboard.
- In the Dashboard list, find the dashboard for which you want to create a backup and choose or > Copy.

For example, if you click Copy for a dashboard named test dashboard, a new dashboard namedtest dashboard_copy is created.

4.4.15.3.3. Manage a dashboard in display mode

This topic describes how to configure a dashboard in display mode. By default, a dashboard shows all charts in display mode. You can perform multiple operations on a dashboard in display mode.

Specify a time range for a dashboard

By default, the time range that you specify for a dashboard applies to all charts on the dashboard. After you specify a time range for a dashboard, all charts on the dashboard display the query and analysis results of the time range. For information about how to specify a time range for a single chart, see Specify a time range for a chart.

1. Log on to the Log Service console

- 2. In the Projects section, click the name of the project in which you want to manage a dashboard.
- 3. In the left-side navigation pane, click the Dashboard icon.
- 4. In the Dashboard list, click the dashboard that you want to manage.
- 5. Click Time Range to specify a time range.
 - Log Service supports the following types of time ranges:
 - Relative: queries log data that is generated within a time range that ends with the current time, such as the previous 1, 5, or 15 minutes. The time
 range is accurate to seconds. For example, if the current time is 19:20:31 and you select 1Hour(Relative) as the time range, the charts on the
 dashboard display the log data that is generated from 18:20:31 to 19:20:31.
 - Time Frame: queries log data that is generated within a time range that ends with the current time, such as the previous 1 or 15 minutes. The time
 range is accurate to minutes, hours, or days. For example, if the current time is 19:20:31 and you select 1Hour(Time Frame) as the time range, the
 charts on the dashboard display the log data that is generated from 18:00:00 to 19:00:00.
 - Custom: queries log data that is generated within a custom time range.
- 6. Mover the pointer over the Time Range button to confirm the specified time range.

Enter the edit mode

In the Dashboard list, click the dashboard that you want to modify. On the page that appears, click **Edit** to enter the edit mode. In edit mode, you can perform multiple operations on the dashboard and the charts on the dashboard. For more information, see Manage a dashboard in edit mode.

Configure an alert rule

Configure a refresh method

In the upper-right corner of the chart, click and select -> Create Alert to create an alert rule for the charts on the dashboard. For more information, see Configure an alert rule.

see configure an alere fuie.

You can manually refresh a dashboard or select an interval to automatically refresh the dashboard.

- In the upper-right corner of the dashboard page, choose **Refresh > Once**. The dashboard is immediately refreshed.
- In the upper-right corner of the dashboard page, choose **Refresh** > **Auto Refresh** and select an interval at which the dashboard is automatically refreshed.

The interval can be 15 seconds, 60 seconds, 5 minutes, or 15 minutes.

```
⑦ Note
If your browser is inactive, the dashboard may not refresh at the specified interval as expected.
```

Share a dashboard

In the upper-right corner of the dashboard page, click **Share** to copy the link of the dashboard to the clipboard. Then, you can send the link to authorized users. The shared dashboard page uses the settings of the dashboard at the point in time when you share the dashboard. The settings include the time range of charts and the display format of chart titles.

? Note

Before you share a dashboard with other users, you must grant the read permissions to the users.

Display a dashboard in full screen

In the upper-right corner of the dashboard page, click **Full Screen**. Then, the dashboard enters the full-screen mode. This mode is suitable for scenarios such as presentations and reporting.

Select a display format for chart titles

In the upper-right corner of the dashboard page, click Title Configuration to select a display format for chart titles. Valid values:

- Single-line Title and Time Display
- Title Only
- Time Only

Reset the time range

In the upper-right corner of the dashboard page, click **Reset Time** to restore the saved time range of all charts on the dashboard. You can use this feature to restore time settings.

Configure charts

You can select a chart and perform the following operations on the chart.

```
? Note
```

Different types of charts on a dashboard can display different information. You cannot view the analysis details of non-statistical charts such as custom charts and Markdown charts.

View analysis details

Find the chart whose details you want to view, click the ; icon and select **View Analysis Details**. On the page that appears, you can view the query statement and the properties of the chart.

Specify a time range for a chart

Find the chart that you want to manage, click the ; icon and select Select Time Range to specify a time range for the chart.

• Configure an alert rule for a chart

Find the chart for which you want to configure an alert rule, click the ; icon and select **Create Alert** to create an alert rule for the chart. For more information, see Configure an alert rule.

Download log data

Find the chart whose log data you want to download, click the \ddagger icon and select **Download Chart Data**. The log data that is returned by the query statement of the chart within the current time range is downloaded in a comma-separated values (CSV) file.

Check whether a drill-down event is configured for a chart

Find the chart that you want to check, click the : icon and move the pointer over the sicon to check whether a drill-down event is configured for the chart. If the icon is red, a drill-down event is configured for the chart.

Preview the query statement of a chart

Find the chart whose query statement you want to preview, click the \ddagger icon, and then click the \circledcirc icon. In the Preview Query Statement dialog box, you can view the query statement of the chart.

4.4.15.3.4. Manage a dashboard in edit mode

You can manage a dashboard in edit mode. For example, you can add chart elements, adjust chart layouts, edit charts, and change the dashboard name.

Add chart elements

- 1. In the Projects section, click the project in which you want to modify a dashboard.
- 2. In the left-side navigation pane, click the $\rm equation$ icon.
- 3. In the Dashboard list, click the dashboard that you want to manage.
- 4. In the upper-right corner of the dashboard page, click **Edit**.

You can add the following chart elements on a dashboard in edit mode.

() Important If you modify a dashboard in edit mode, you must save the modifications before the modifications can take effect. To save modifications, click **Save** in the upper-right corner of the dashboard page.

• Rectangles and diamonds

Drag the rectangular icon or the diamond icon to a position. Then, double-click the icon and enter text. You can also modify the text properties and the border properties of rectangles and diamonds.

Common icons

Log Service allows you to display common icons on a dashboard page. You can drag an icon to a position.

• Text

Drag the text icon to a position. Then, double-click the text box and enter text. You can also modify the properties of the text. The properties include the font size, font style, alignment, and font color.

Markdown chart

Drag the Markdown icon to a position. Then, double-click the text box and insert elements such as text, charts, and videos. For more information, see Manage a Markdown chart.

• Filter

Click the filter icon to add a filter. For more information, see Add a filter.

After you add a filter to a dashboard, you can use the filter to refine search results or replace placeholder variables in query statements.

Custom SVG
 Click the SVG icon. In the Customize SVG dialog box, click the box or drag a Scalable Vector Graphics (SVG) file to the box to upload the file.

⑦ Note The size of an SVG file cannot exceed 10 KB.

Custom image's HTTP link

Click the Customize image's HTTP link icon in the menu bar. On the page that appears, enter the HTTP link of an image and click OK.

Adjust chart layouts

On a dashboard in edit mode, all charts and chart elements are displayed on a canvas. You can drag and scale each chart. The width of the canvas cannot exceed the width of your browser. The height of the canvas is unlimited and is measured in pixels.

On the canvas, you can perform the following operations:

- Adjust the position of a chart.
 - $\circ~$ You can drag a chart to a position.
 - $\circ~$ You can select a chart and set the ${\bm L}$ and ${\bm T}$ parameters to adjust the chart position.
- · Adjust the width and height of a chart.
 - You can select a chart and drag the lower-right corner of the chart to resize the chart.
 - You can select a chart and set the **W** and **H** parameters to resize the chart.
- Add lines to connect charts.

You can add a directional line between two charts. When you adjust the position or size of the charts, the line automatically moves to show the relative position between the two charts.

Configure chart levels.

You can select a chart and click the Move Layer to Top icon or the Move Layer to Down icon in the menu bar to move the chart to the upper part or lower part of the dashboard.

Configure charts

You can modify, copy, and delete a chart on a dashboard in edit mode.

- Modify the query statement, properties, data source, and interactive behavior for a chart.
- i. In the upper-right corner of the dashboard page, click Edit to enter the edit mode. Find the chart that you want to modify and choose + > Edit.
- ii. Modify the query statement, properties, data source, and interactive behavior for the chart.
- For information about how to configure interactive behavior for a chart, see Configure a drill-down event.
- iii. Click **Preview** to check the configuration results.
- iv. Click OK.
- v. In the upper-right corner of the dashboard page, click **Save**.
- Create a copy of a chart. The copy uses the same configurations as the chart.
- i. In the upper-right corner of the dashboard page, click Edit to enter the edit mode. Find the chart that you want to copy and choose , > Copy.
- ii. Drag the copy to a position. Then, specify the margins and size of the copy.
- iii. In the upper-right corner of the dashboard page, click **Save**.
- Delete a chart.
- i. In the upper-right corner of the dashboard page, click Edit to enter the edit mode. Find the chart that you want to delete and choose , > Delete.
- ii. In the upper-right corner of the dashboard page, click Save.

4.4.15.3.5. Configure a drill-down event

Log Service allows you to configure a drill-down event for a chart to obtain more details in analysis results. This topic describes how to configure a drilldown event in the Log Service console.

Prerequisites

- The indexing feature is enabled and indexes are configured. For more information, see Configure indexes.
- A Logstore is created. This prerequisite must be met if you want to configure a drill-down event to open a Logstore. For more information, see Create a Logstore.
- A saved search is created. This prerequisite must be met if you want to configure a drill-down event to open a saved search. For more information, see Saved search.

Placeholder variables are configured in the query statement of the saved search. This prerequisite must be met if you want to configure variables. For more information, see Configure a placeholder variable.

• A dashboard is created. This prerequisite must be met if you want to configure a drill-down event to open a dashboard. For more information, see Create a dashboard.

Placeholder variables are configured in the related chart on the dashboard. This prerequisite must be met if you want to configure variables. For more information, see Configure a placeholder variable.

• If you want to configure a drill-down event to open a custom HTTP URL, you must create the HTTP URL.

Background information

Drilling is required for data analysis. This feature allows you to analyze data in a fine-grained manner or coarse-grained manner. Drilling includes roll-up and drill-down. Drill-down allows you to obtain more details in analysis results. This way, you extract more value from data and make better decisions for your business.

Procedure

- 1. Log on to the Log Service console.
- 2. Click the name of the project that you want to manage.
- 3. Click the icon next to the name of the Logstore in which you want to query and analyze data, and then select Search & Analysis.
- 4. Enter a query statement in the search box, specify a time range, and then click Search & Analyze.
- 5. On the **Graph** tab, select a chart type. On the Properties tab, configure the parameters.
- For information about the parameters of a chart, see Chart configurations.

6. Click the Interactive Behavior tab. On this tab, configure a drill-down event for the chart.

You can set the Event Action parameter to Disable, Open Logstore, Open Saved Search, Open Dashboard, Open Dashboard, or Custom HTTP Link. • **Disable**: disables the drill-down feature.

• Disable: disables the drill-down reature

• **Open Logstore**: configures the drill-down event to open a Logstore. The following table describes the parameters that you can configure if you set the Event Action parameter to Open Logstore.

Parameter	Description		
Open in New Tab	If you turn on this switch, the Logstore that you specify is opened on a new tab when the drill-down event is triggered.		
Select Logstore	The name of the Logstore to which you want to be redirected. When a drill-down event is triggered, you are redirected to the Search & Analysis page of the Logstore.		
	The time range. The system queries the data that is generated within the time range. Valid values:		
	 Default: queries data in the Logstore to which you are redirected based on the default time range. The default time range is 15 minutes (relative) and accurate to seconds. 		
Time Range	 Inherit table time: queries data in the Logstore to which you are redirected based on the time range specified for the chart when the drill-down event is triggered. 		
	 Relative: queries data in the Logstore to which you are redirected based on the time range that you specify. The time range is accurate to seconds. 		
	 Time Frame: queries data in the Logstore to which you are redirected based on the time range that you specify. The time range is accurate to minutes, hours, or days. 		
Inherit Filtering Conditions	If you turn on Inherit Filtering Conditions , the filter conditions that are added to the dashboard are synchronized to the Search & Analysis page of the Logstore to which you are redirected when the drill-down event is triggered. The filter conditions are added to the start of the query statement by using the AND operator.		
Filter	On the Filter tab, you can enter a filter statement in the Filter Statement field. When the drill-down event is triggered, the filter statement is synchronized to the Search & Analysis page of the Logstore to which you are redirected. The filter statement is added to the start of the query statement by using the AND operator.		
	The filter statement can contain fields that you specify in the Optional Parameter Fields field.		

• **Open Saved Search**: configures the drill-down event to open a saved search. The following table describes the parameters that you can configure if you set the Event Action parameter to Open Saved Search.

Parameter	Description	
Open in New Tab	If you turn on this switch, the saved search that you specify is opened on a new tab when the drill-down event is triggered.	
Select Saved Search	The name of the saved search to which you want to be redirected. When a drill-down event is triggered, you are redirected to the page of the saved search.	
Time Range	 The time range for the saved search. Valid values: Default: queries data by using the saved search based on the default time range. The default time range is 15 minutes (relative) and accurate to seconds. Inherit table time: queries data by using the saved search based on the time range specified for the chart when the drill-down event is triggered. Relative: queries data by using the saved search based on the time range that you specify. The time range is accurate to seconds. Time Frame: queries data by using the saved search based on the time range that you specify. The time range is accurate to minutes, hours, or days. 	
Inherit Filtering Conditions	If you turn on Inherit Filtering Conditions , the filter conditions that are added to the dashboard are synchronized to the saved search that you want to execute when the drill-down event is triggered. The filter conditions are added to the start of the saved search by using the AND operator.	
Inherit Variables	If you turn on Inherit Variables and the variable that you configure on the dashboard is the same as the variable in the saved search, the variable value on the dashboard replaces the variable in the saved search. ⑦ Note If you want to inherit variables, you must configure a placeholder variable in the saved search.	
Filter	On the Filter tab, you can enter a filter statement in the Filter Statement field. When the drill-down event is triggered, the filter statement is added to the start of the saved search by using the AND operator. The filter statement can contain fields that you specify in the Optional Parameter Fields field.	
Variable	 Log Service allows you to modify a saved search by using variables. If you configure a variable that is the same as the variable in the saved search, the variable value that you click to trigger the drill-down event replaces the variable in the saved search. You can add variables on the Variable tab. ⑦ Note If you want to configure a variable, you must configure a placeholder variable for the saved search to which you want to be redirected. You can add up to five dynamic variables and up to five static variables. Dynamic variables Variable: the name of the variable. Variable: the name of the variable. Static variables Variable: the name of the variable. Variable: the name of the variable. Variable: the name of the variable. Variable: the name of the variable. Variable: the name of the variable.	

• Open Dashboard: configures the drill-down event to open a dashboard. The following table describes the parameters that you can configure if

you set the Event Action parameter to Open Dashboard.

Parameter	Description		
Open in New Tab	If you turn on this switch, the dashboard that you specify is opened on a new tab when the drill-down event is triggered.		
Select Dashboard	The name of the dashboard to which you want to be redirected. When a drill-down event is triggered, you are redirected to the page of the dashboard.		
Time Range	 The time range to query data for the dashboard. Valid values: Default: queries data for the dashboard to which you are redirected based on the default time range. The default time range is 15 minutes (relative) and accurate to seconds. Inherit table time: queries data for the dashboard to which you are redirected based on the time range specified for the chart when the drill-down event is triggered. Relative: queries data for the dashboard to which you are redirected based on the time range that you specify. The time range is accurate to seconds. Time Frame: queries data for the dashboard to which you are redirected based on the time range that you specify. The time range is accurate to minutes, hours, or days. 		
Inherit Filtering Conditions	If you turn on Inherit Filtering Conditions , the filter conditions that are added to the current dashboard are synchronized to the dashboard to which you are redirected when the drill-down event is triggered.		
Inherit Variables	If you turn on Inherit Variables , the variables that you configure on the current dashboard are synchronized to the dashboard to which you are redirected.		
Filter	On the Filter tab, you can enter a filter statement in the Filter Statement field. When the drill-down event is triggered, the filter statement is synchronized to the dashboard to which you are redirected. The filter statement can contain fields that you specify in the Optional Parameter Fields field.		
Variable	 The variables that you configure are synchronized to the dashboard to which you are redirected when the drill-down event is triggered. You can add variables on the Variable tab. Note If you want to configure a variable for the chart, you must configure a placeholder variable for the chart on the dashboard to which you are redirected. You can add up to five dynamic variables and up to five static variables. Dynamic variables Variable: the name of the variable. Variable Value Column: the column in which the variable values are located. The values are dynamically synchronized to the dashboard to which you are redirected. Static variables Variable: the name of the variable. Variable: the name of the variable. Variable: the name of the variable. Variable: the name of the variable. Variable: the name of the variable. 		

 $\circ~$ Custom HTTP Link: configures the drill-down event to open a custom HTTP URL.

The path in the custom HTTP URL is the path of the file that you want to access. You can add an optional parameter to the path. If you click a variable value on a chart to trigger the drill-down event, the parameter is replaced by the value, and you are redirected to the custom HTTP URL.

Parameter	Description		
Enter Link	The URL to which you want to be redirected.		
Use System Variables	If you turn on Use System Variables , you can insert the following variables that are provided by Log Service to the HTTP URL: \${sls_project}, \${sls_dashboard_title}, \${sls_chart_name} , \${sls_chart_title}, \${sls_region}, \${sls_start_time}, \${sls_end_time}, \${sis_realUid}, and \${sls_aliUid} .		
Transcoding	If you turn on Transcoding , the custom HTTP URL is encoded.		
Optional Parameter Fields	If you add an optional parameter to the path, the parameter is replaced by the value that you click to trigger the drill-down event.		

7. Click Add to New Dashboard.

8. In the dialog box that appears, specify a dashboard name and a chart name, and then click OK.

Example

This section provides an example on how to store NGINX access logs in a Logstore named accesslog and how to create two dashboards named RequestMethod and destination_drilldown for drill-down analysis. Before you perform drill-down analysis, add a table of request methods to the RequestMethod dashboard, and configure a drill-down event for the table to open the destination_drilldown dashboard. Then, add a line chart to the destination_drilldown dashboard. The line chart displays the trend of page views (PVs) over a specified period of time. After you configure the settings, you can click a request method on the RequestMethod dashboard. Then, you are redirected to the destination_drilldown dashboard on which can view the trend of PVs over a specified period of time.

1. Create a dsahboard named destination_drilldown.

Before you configure a drill-down event for the table of request methods, create a dashboard to which you want to be redirected and add a line chart to the dashboard. The line chart displays the trend of PVs over a specified period of time. You need to configure the following settings. For more information, see Create a dashboard.

Specify a query statement.
 The query statement queries logs by request type. You can view the trend of PVs over a specified period of time.

request_method: * | SELECT date_format(date_trunc('minute', __time_), '%H:%i:%s') AS time, COUNT(1) AS PV GROUP BY time ORDER BY time

• Configure a placeholder variable.

Specify the asterisk (*) to generate a placeholder variable and set the variable name to **method**.



 Configure a drill-down event for the table of request methods and add the table to the RequestMethod dashboard. You need to configure the following settings. For more information, see Procedure.

• Specify a query statement.

The query statement queries the logs that are generated for each request method among the NGINX access logs.

*|SELECT request_method, COUNT(1) AS c GROUP BY request_method ORDER BY c DESC LIMIT 10

Select a chart type.

In this example, a table is selected.

- Configure a drill-down event for the table.
 - Configure a drill-down event for the **request_method** column in the table.
 - Set the Select Dashboard parameter to destination_drilldown.
 - Set the Variable parameter to method.

Drilldown Configu	rations	Event Action	
request_method	Configure×	Open Dashboard	\sim
		Select Dashboard:	
		destination_drilldown	\sim
		Time Range:	
		Inherit table time	\sim
		Inherit Filters:	
		Variable	
		method X	

3. View drill-down results.

On the RequestMethod dashboard, click **GET**. You are redirected to the destination_drilldown dashboard. The asterisk (*) in the query statement is replaced by the value **GET**. The trend of PVs for GET requests over a specified period of time is displayed in a line chart.

request_method $\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$		c ¢Q
GFT		5452
Custom Event		1417
		1337
DELETE		562
HEAD		31



4.4.15.3.6. Add a filter

You can add a filter to a dashboard. Then, you can use the filter to refine query results or replace placeholder variables with specific values. This topic describes how to add a filter to a dashboard.

Prerequisites

- Log data is collected. For more information, see Data collection overview.
- Indexes are configured. For more information, see Configure indexes.
- Charts are added to a dashboard. For more information, see Add a chart to a dashboard.

① Important If you set the filter type to Replace Variable, you must configure placeholder variables for the charts on the dashboard.

Background information

A filter is used to modify query statements or replace placeholder variables for all charts on a dashboard. Each chart displays the query and analysis result of a query statement, which is in the [search query] [sql query] format. After you add a filter to a dashboard, the filter condition or variables that you specify for the filter apply to the query statement that corresponds to each chart on the dashboard. The following types of filters are supported:

· Filter: uses key-value pairs as a filter condition.

The filter condition is added to the start of a query statement by using the **AND** or **NOT** operator. For example, the **Key: Value AND [search query] | [sql query]** statement queries logs that contain **Key:Value** in the query result of the [search query] | [sql query] statement. For the Filter type, you can select or enter multiple key-value pairs. If you specify multiple key-value pairs, the logical OR operator is used between the pairs.

Replace Variable: uses a variable and the value of the variable.
 If the variable that you specify for the filter is configured for existing charts on the dashboard, the variable in the query statement of each chart is automatically replaced by the variable value that you specify for the filter. This applies to all charts for which the same variable is configured.

Procedure

- 1. Log on to the Log Service console.
- 2. Click the name of the project that you want to manage.
- 3. In the left-side navigation pane, click the Dashboard icon.
- 4. In the Dashboard list, click the dashboard that you want to manage.
- 5. In the upper-right corner of the dashboard page, click **Edit**.
- 6. Click the jicon and configure the parameters. Then, click **OK**. The following table describes the parameters.

Parameter	Description	
Filter Name	The name of the filter.	
Display Settings	 Valid values: Title: specifies whether to add a title for the filter. You can turn on Title to add a title for the filter. Border: specifies whether to add borders to the filter. You can turn on Border to add borders to the filter. Background: specifies whether to add a white background to the filter. You can turn on Background to add a white background to the filter. 	
Туре	 The type of the filter. Filter: uses key-value pairs to filter data. The key-value pairs are used as a filter condition and are added to the start of a query statement by using the AND or NOT operator. By default, the AND operator is used. AND: Key: Value AND [search query] [sql query] NOT: Key: Value NOT [search query] [sql query] You can specify multiple values for the key-value pairs in theStatic List Items field. Replace Variable: specifies a variable and the value of the variable. If the variable that you specify for the filter is configured for exiting charts on the dashboard, the variable in the query statement of each chart is automatically replaced by the variable value that you specify for the filter. You can specify multiple values for the variables in the Static List Items field. 	
Кеу	 If you select Filter, enter the key that you want to use to filter data in theKey field. If you select Replace Variable, enter the variable that you want to use to filter data in theKey field. Note If you select Replace Variable, you must specify a placeholder variable when you add a chart to the dashboard. The placeholder variable must be the same as the variable that you specify in the Key field. 	

Alias	The alias of the key. This parameter is available only when you select Filter .
Global filter	 The parameter is available only when you select Filter. If you want to filter specific values in all fields, turn onGlobal filter. If you want to filter specific values in specified keys, turn offGlobal filter.
Static List Items	The value of the Key field that is used to filter data. You can click the plus sign (+) to add more values for the specified key. If you turn or Select by Default for a value, the value is used to filter data each time you open a dashboard.
Add Dynamic List Item	If you turn on Add Dynamic List Item , dynamic values can be retrieved for Key . Dynamic list items are dynamic values that are retrieved by executing the specified query statement. The values vary based on the time ranges during which the query statement is executed. If you turn on Add Dynamic List Item , you must configure the following parameters:
	 Inherit Filtering Conditions: If you turn on Inherit Filtering Conditions, the filter condition on the dashboard is added before the query statement.
	• Query statement: Enter a query statement and specify a time range.
	Dynamic List Item Preview: Preview query results.

4.4.15.3.7. Manage a Markdown chart

Log Service allows you to add a Markdown chart to a dashboard. In the Markdown chart, you can insert images, links, videos, and other elements to make your dashboard page more intuitive to use.

Background information

You can add multiple analysis charts to a dashboard. This allows you to view multiple analysis results and monitor the status of multiple applications on a single dashboard. You can also add Markdown charts to a dashboard. A Markdown chart is edited by using the Markdown syntax.

You can create different Markdown charts based on your business requirements. Markdown charts can make a dashboard more intuitive to use. You can insert text such as background information, chart description, notes, and extension information into a Markdown chart. You can insert custom images and videos into a Markdown chart. You can insert saved searches or the dashboard links of other projects to redirect to other query pages. You can insert links into a Markdown chart to redirect to the other dashboard pages of the current project. You can also insert an image that corresponds to each link. In addition, you can use a Markdown chart to describe the parameters of analysis charts.



Add a Markdown chart

- 1. Log on to the Log Service console.
- 2. In the Projects section, click the name of the project in which you want to manage a dashboard.
- 3. In the left-side navigation pane, click the right icon.
- 4. In the Dashboard list, click the dashboard that you want to manage.
- 5. In the upper-right corner of the dashboard page, click **Edit**.
- 6. In edit mode, drag the information from the menu bar and drop the icon on a specified position to create a Markdown chart.
- 7. Double-click the Markdown chart.
- 8. In the Markdown Edit dialog box, set the parameters, and then click OK.

Parameter	Description
Chart Name	The name of the Markdown chart.
Show Border	Specifies whether to show the borders of the Markdown chart. You can turn on Show Border to show the borders of the Markdown chart.
Show Title	Specifies whether to show the title of the Markdown chart. You can turn or Show Title to show the title of the Markdown chart.

Show Background	Specifies whether to show the background of the Markdown chart. You can turn on Show Background to show the background of the Markdown chart.
Query Binding	 Specifies whether to associate a query statement with a Markdown chart. You can turn onQuery Binding and associate a query statement with a Markdown chart. Then, dynamic query results are displayed in the Markdown chart. i. Select a Logstore whose data you want to query. ii. Enter a query statement in the search box, specify a time range, and then clickSearch. For more information, see Log search overview.
	③ Note The query results may contain logs that are generated 1 minute earlier or later than the specified time range.
	The first returned log is displayed.
	iii. Click the plus sign next to a field to insert the corresponding query result into the Markdown Content column.
Markdown Content	Enter Markdown content in the Markdown content column on the left. The data preview is displayed in real time in the Show Chart column on the right. You can modify the Markdown content based on the data preview. For more information, see Common Markdown syntax.

9. Click Save.

Modify a Markdown chart

- 1. In the upper-right corner of the dashboard page, click **Edit**.
- Modify the position and size of a Markdown chart Drag the Markdown icon to a position on the dashboard and drag the lower-right corner of the chart to adjust the size of the chart.
- 3. Modify the properties of a Markdown chart
- i. Double-click the Markdown chart that you want to modify.
- ii. In the Markdown Edit dialog box, modify the parameters, and then click OK. You can modify the chart name, display settings, query settings, and Markdown content. For more information, see Add a Markdown chart.

Delete a Markdown chart

- 1. In the upper-right corner of the dashboard page, click Edit.
- 2. Find the Markdown chart that you want to delete and choose > Delete.
- 3. In the upper-right corner of the dashboard page, click Save.

Common Markdown syntax

Heading

- Markdown syntax
 - # Level 1 heading ## Level 2 heading ### Level 3 heading

Result

wd-test

Level 1 title Level 2 title Level 3 title

• Link Markdown syntax

Contents

```
[Test](https://www.alibabacloud.com/)
```

- Image
 - Markdown syntax

<div align=center>

![Alt txt][id]

With a reference later in the document defining the URL location

[id]: https://octodex.github.com/images/dojocat.jpg "The Dojocat"

• Preview

	Image			
	With a reference later in the document defining the URL location			
• Sp °	becial tag Markdown syntax			
	Advertisement :)			
	<pre>=some mark== `some code` > Classic markup: :wink: :crush: :cry: >> Shortcuts (emoticons): :-) 8-) ;)</pre>	:tear:	:laughing:	:
	This is bold text			
	This is italic text			
0	Result			
	Code			
	Advertisement 🚭			
	some mark some code			
	Classic markup: 😳 :crush: 😳 :tear: 🗟 😂			
	Shortcuts (emoticons): 🍚 🕏 😳			
	This is bold text			
	This is italic text			

4.5. Data transformation

4.5.1. Data transformation overview

Data transformation is a fully managed feature that provides high availability and scalability in Log Service. You can use the data transformation feature to standardize, enrich, transfer, mask, and filter data.

yum:

Transformation process

- Log Service transforms data in the following steps:
- 1. A consumer group reads data from a source Logstore.
- 2. Log Service transforms each data entry based on a transformation rule.
- 3. Log Service writes the transformed data to a destination Logstore.
- After data is transformed, you can view the results in the destination Logstore.

Features

You can use the data transformation feature to standardize, enrich, transfer, mask, and filter data.

- Data standardization: Log Service can extract fields from logs in different formats and convert the log formats to obtain structured data for stream
 processing and computing in data warehouses.
- Data enrichment: Log Service can join the fields of logs and dimension tables to link logs with dimension information. For example, Log Service can join the fields of order logs and a user information table. This facilitates data analysis.
- Data transfer: Log Service can transfer logs from regions outside the Chinese mainland to a central region by using the global acceleration feature. This helps you manage global logs in a centralized manner.
- Data masking: Log Service can mask sensitive information in data, such as passwords, mobile phone numbers, and addresses.
- Data filtering: Log Service can filter logs to obtain key service logs. This helps further analysis.

Scenarios

• Data standardization: Log data is read from a source Logstore, transformed, and then written to a destination Logstore.



• Data transfer: Log data is read from a source Logstore, transformed, and then written to multiple destination Logstores.



• Multi-source data aggregation: Log data is read from multiple source Logstores, transformed, and then written to a destination Logstore.



Transformation syntax

The domain-specific language (DSL) for Log Service provides more than 200 built-in functions and more than 400 regular expressions. You can use the DSL for Log Service to create user-defined functions (UDFs). For more information, see Syntax overview.

Benefits

- Allows you to use the DSL for Log Service to orchestrate functions based on your business requirements. You can use the orchestrated functions to filter, standardize, enrich, transfer, and mask data.
- Processes data in real time and allows you to view data within seconds. The feature scales the computing capability based on the size of data and provides a high throughput.
- Is suitable for log analysis scenarios and provides out-of-the-box functions.
- Provides real-time dashboards, exception logs, and alert integration.
- Offers a fully-managed and maintenance-free service that can be integrated with big data services of Alibaba Cloud and open source ecosystems.

4.5.2. Terms

This topic introduces the terms that are related to the data transformation feature.

Terms

• ETL

Extract, transform, and load (ETL) is a process during which data is extracted from business systems, cleansed, transformed, and loaded. This process unifies and standardizes data from different sources. Log Service can load data from a source Logstore, transform data, and then write transformed data to destination Logstores. Log Service can also load data from Object Storage Service (OSS) buckets, ApsaraDB RDS instances, or other Logstores.

event, data, and log

In data transformation, events and data are represented by logs. For example, the event time is equivalent to the log time, and the drop_event_fields function discards log fields.

log time

The log time indicates the point in time at which an event occurs. The log time is also known as the event time. The log time is indicated by the reserved field __time__ in Log Service. The value of this field is extracted from the time information in logs. The value is a UNIX timestamp. It is the number of seconds that have elapsed since 00:00:00 UTC, Thursday, January 1, 1970. Data type: integer. Unit: seconds.

log receiving time

The log receiving time indicates the point in time at which a log is received by a server of Log Service. By default, this time is not saved in logs. However, if you turn on Log Public IP for a Logstore, this time is recorded in the log tag field <u>receive_time</u>. In the data transformation process, the complete name of this field is <u>tag: receive_time</u>. The value is a UNIX timestamp. It is the number of seconds that have elapsed since 00:00:00 UTC, Thursday, January 1, 1970. Data type: integer. Unit: seconds.

ONote In most scenarios, logs are sent to Log Service in real time, and the log time is the same as the log receiving time. If you import historical logs, the log time is different from the log receiving time. For example, if you import logs generated during the last 30 days by using an SDK, the log receiving time is the current time and is different from the log time.

- tag
- Logs have tags. Each tag field is prefixed with __tag_: . Log Service supports the following types of tags:
- Custom tags: the tags that you add when you call the PutLogs operation to write data.
- $\circ~$ System tags: the tags that are added by Log Service, including <code>__client_ip__</code> and <code>__receive_time__</code> .

Configuration-related terms

source Logstore

The data transformation feature reads data from a source Logstore for transformation.

You can configure only one source Logstore for a data transformation job. However, you can configure the same source Logstore for different data transformation jobs.

destination Logstore

The data transformation feature writes transformed data to destination Logstores.

You can configure one or more destination Logstores for a data transformation job. Data can be written to destination Logstores in static or dynamic mode.

SLS DSL

The domain-specific language (DSL) for Log Service is a Python-compatible scripting language, and is used for data transformation in Log Service. The DSL for Log Service is built on top of Python. The DSL provides more than 200 built-in functions to simplify common data transformation jobs. The DSL also allows you to use custom Python extensions. For more information, see Language overview.

• transformation rule

A transformation rule is a data transformation script that is orchestrated by using the DSL for Log Service.

· data transformation job

A data transformation job is the minimum scheduling unit of data transformation. You must configure a source Logstore, one or more destination Logstores, a transformation rule, a transformation time range, and other parameters for a data transformation job.

Rule-related terms

resource

Resources refer to third-party data sources that are referenced during data transformation. The data sources include but are not limited to onpremises resources, Object Storage Service (OSS), ApsaraDB RDS, and Logstores other than the source and destination Logstores. The resources may be referenced to enrich data. For more information, see Resource functions.

dimension table

A dimension table contains dimension information that can be used to enrich data. A dimension table is an external table. For example, a dimension table can contain the information of users, products, and geographical locations of a company. In most scenarios, dimension tables are included in resources and may be dynamically updated.

enrichment or mapping

If the information contained in a log cannot meet your requirements, you can map one or more fields in the log by using a dimension table to obtain more information. This process is called enrichment or mapping.

For example, a request log contains the status field that specifies the HTTP status code. You can map the field to the status_desc field to obtain the HTTP status description by using the following table.

Before enrichment	After enrichment
status	status_desc
200	Success
300	Redirect
400	Permission error
500	Server error

If a source log contains the user_id field, you can map the field by using a dimension table that contains account details to obtain more information. For example, you can obtain the user name, gender, registration time, and email address for each user ID. Then, you can add the information to the source log and write the log to the destination Logstores. For more information, see <u>Mapping and enrichment functions</u>.

event splitting

If a log contains multiple pieces of information, the log can be split into multiple logs. This process is called event splitting.

For example, a log contains the following information:

```
_time_: 1231245
__topic: "win_logon_log"
content:
[ {
    "source": "192.0.2.1",
    "action": "login",
    "result": "pass"
},{
    "source": "192.0.2.2",
    "dest": "192.0.2.1"
    "action": "logout",
    "result": "pass"
}
]
```

The log can be split into two logs.

```
__time__: 1231245
__topic: "win_logon_log"
content:
{
  "source": "192.0.2.1",
  "dest": "192.0.2.1"
  "action": "login",
 "result": "pass"
}
__time__: 1231245
____topic: "win_logon_log"
content:
{
  "source": "192.0.2.2",
 "dest": "192.0.2.1"
 "action": "logout",
  "result": "pass"
```

```
)
• GROK
```

Grok uses patterns to replace complex regular expressions.

· content capturing by using a regular expression

You can use a regular expression to capture specified content in a field and include the content in a new field.

For example, the function e_regex("content", "(?P<email>[a-zA-Z][a-zA-

4.5.3. Data transformation basics

The data transformation feature uses consumer groups to consume log data and uses transformation rules to transform log data. More than 200 built-in functions are available for you to orchestrate transformation rules. This topic describes how log data consumption is scheduled during data transformation and how the rules engine for data transformation works.

Scheduling basics

The data transformation feature of Log Service uses a consumer group to consume log data from the source Logstore in streaming mode, transforms each log based on the specified transformation rule, and then writes the transformed log data to the destination Logstore.



• Scheduling mechanism

For each transformation rule, the data transformation scheduler starts one or more running instances. Each running instance behaves as a consumer to consume data from one or more shards of the source Logstore. The scheduler determines the number of concurrent running instances based on the memory and CPU resources that are used by running instances. The maximum number of running instances that the scheduler can start is the same as the number of shards in the source Logstore.

• Running instance

Running instances read source log data from the shards that are allocated to them based on your configurations. The data is transformed based on the transformation rule and is then written to the destination Logstore. You can configure transformation rules to enrich log data by using external resources. Based on the consumer group mechanism, running instances record data consumption checkpoints in shards. The checkpoints are useful if consumption is unexpectedly interrupted. After an interruption ends, running instances can continue to consume data from the last checkpoint.

Job termination

- By default, if you do not specify the end time of a data transformation job, running instances do not exit and the job does not stop.
- If you specify the end time of a data transformation job, running instances consume log data until the end time. When the job reaches the end time, the instances exit and the job stops.
- By default, if a job is stopped and then restarted, running instances continue to consume data from the last recorded checkpoint.

Rules engine basics: basic operations

You can use the built-in functions that are written in the domain specific language (DSL) for Log Service to orchestrate transformation rules. Each function is a transformation step. The rules engine calls the functions of a transformation rule in sequence. For example, the following transformation rule is orchestrated by using four functions. The functions are four steps that are used to transform data:

- e_set("log_type", "access_log")
- e_drop_fields("__action")
- e_if(e_search("ret: pass"), e_set("result", "pass"))
- e_if(e_search("ret: unknown"), DROP)

The following figure shows the transformation logic.



Basic logic

The rules engine calls each function that is defined in the rule in sequence. Each function processes and modifies each log, and returns a transformed log.

Condition evaluation

You can specify conditions in steps. If a log does not meet a condition in a step, this step is skipped for the log.

For example, the <code>e_if(e_search("ret: pass"), e_set("result", "pass"))</code> function first checks whether the value of the <code>ret</code> field in a log contains pass. If no, this step is skipped for the log. If yes, the function sets the value of the <code>result</code> field in the log to pass.

• Transformation termination

If a function does not return a transformed log, the log is discarded.

For example, the e_if(e_search("ret: unknown"), DROP) function discards a log in which the value of the ret field is unknown. After the log is discarded, the rules engine no longer calls subsequent functions to transform this log, and starts to transform the next log.

Rules engine basics: data output, duplication, and splitting

The rules engine also supports log output, duplication, and splitting. For example, the following transformation rule is orchestrated by using four functions. The functions are four steps that are used to transform data:

- e_coutput("archive_Logstore"))
- e_split("log_type")
- e_if(e_search("log_type: alert"), e_output("alert_Logstore"))
- e_set("result", "pass")

The following example is a sample log to be transformed:

log_type: access,alert

content: admin login to database.

The following figure shows the transformation logic.



Log output

Log output can be considered as a special way to stop transforming a log. As shown in the preceding figure, if the value of the log_type field in a log is alert, the e_output("alert_Logstore") function in step 3 is called to write the log to the specified destination Logstore. Then the log is discarded and the subsequent function is not called.

Log duplication and output

The e_coutput function duplicates a log and writes the duplicated log to the specified destination Logstore. Then, the rules engine continues to call subsequent functions to transform the log. As shown in the preceding figure, logs that are duplicated in step 1 are written to the destination Logstore named archive_Logstore.

Log splitting for parallel processing

If the values of the \log_{type} field in a log are access and alert, the $e_{split}("log_type")$ function in step 2 is called to split the log into two logs. The two logs are the same except the value of the \log_{type} field. The value of the field is access in a log and is alert in the other.

The logs that are generated after splitting are processed in the subsequent steps.

4.5.4. Limits

This topic describes the limits on data transformation in Log Service.

Job configuration

Number of jobs

Item

Description

You can create a maximum of 100 data transformation jobs in a project.

() **Important** When a data transformation job is stopped or complete, the job still consumes the job quota. To prevent the quota from being consumed by the data transformation jobs that are stopped or complete, we recommend that you delete the jobs that you no longer use. For more information, see Manage a data transformation job.

To increase the quota, submit a ticket.

Dependency of a consumer group in a source Logstore	The running of a data transformation job depends on a consumer group in the source Logstore. When a data transformation job is running, do notdelete or reset the consumption checkpoint for the consumer group on which the job depends. If you perform the delete or reset operation, the job consumes data again from the start time that you specify, and duplicate data exists in the result. ① Important The data consumption progress of a job in a shard is updated to the consumer group on which the job depends at regular intervals. This optimizes the efficiency of data transformation. However, the result of the GetCheckPoint operation on the consumer group cannot indicate the latest data transformation progress. For more information, see Data transformation basics, Terms, and Consumer group-related API operations in Log Service Developer Guide.
Number of consumer groups in a source Logstore	You can create a maximum of 30 consumer groups in a Logstore. This way, you can create a maximum of 30 data transformation jobs in a source Logstore. If you create more than 30 consumer groups, the data transformation jobs cannot run as expected after the jobs are started. The operational logs of the jobs record error information. () Important When a data transformation job is stopped or complete, Log Service does not automatically delete the consumer group on which the job depends. To reduce invalid consumer groups, we recommend that you delete the data transformation jobs that are stopped or complete and you no longer use. For more information, see Manage a data transformation job.
Modification of time ranges for jobs	 If you modify the time range for a running job, the job starts consumption from the start time that you specify and consumes all data that is generated in the newly specified time range. i. If you want a job to consume data that is generated within a longer time range, we recommend that you create another job to expand the time range instead of prolonging the time range of the existing job. ii. If you want a job to consume data that is generated within a shorter time range, we recommend that you delete the data that is written to the storage destinations and then shorten the time range of the existing job to prevent data duplication. The data that is written to the storage destinations is not automatically deleted.
Number of storage destinations	You can configure a maximum of 20 independent static storage destinations for a data transformation job. A maximum of 200 projects and 200 Logstores can be dynamically specified in data transformation code. If one of the preceding limits is exceeded, the data that is written to a different storage destination other than the allowed 20 storage destinations is discarded.

Data transformation

Item	Description
Quick preview	 The quick preview feature of data transformation is used to debug data transformation code. The feature has the following limits: Connections to external resources such as ApsaraDB RDS, Object Storage Service (OSS), and Log Service are not supported. You can specify custom test data for a dimension table. A single request can obtain no more than 1 MB of test data from a source table or a dimension table. If the size of the data exceeds 1 MB, an error is returned. A maximum of the first 100 logs can be returned for a single request. The advanced preview feature does not have the limits.
Runtime concurrency	 The number of readwrite shards in a source Logstore specifies the maximum number of data transformation jobs that can concurrently run. For more information, see Data transformation basics. For more information about the limits on shards, seeData read and write. For information about how tosplit a shard for a Logstore, see Manage shards. Important If the number of data transformation jobs that can concurrently run does not meet the requirements, automatic sharding is not triggered for the source Logstore, and you must manually split a shard of the source Logstore to increase the number of data transformation jobs that can concurrently run. For more information about automatic sharding, see Manage shards. For data that is written after the shard is split, the maximum number of data transformation jobs that can concurrently run equals the number of readwrite shards that are available in the source Logstore when the data is written.
Data load of a concurrent unit	The data load of a concurrent unit in a data transformation job is determined by the amount of data that is consumed by the job from a shard of the source Logstore. If the data in the source Logstore is unevenly distributed among shards, the data load of a concurrent unit in a data transformation job may be significantly heavier, and the concurrent unit is considered a hot concurrent unit. As a result, the processing of data in some shards is delayed. If data is written to the source Logstore in KeyHash mode, we recommend that you appropriately allocate hash keys and shards to minimize unbalanced data distribution. For more information about data write, see PutLogs in Log Service Developer Guide.
Memory usage	The memory usage threshold for a concurrent unit in a data transformation job is 6 GB. If the memory occupied by a concurrent unit exceeds 6 GB, the performance of the job is limited, and processing latency exists. The memory occupied by a concurrent unit exceeds the threshold when a large number of log groups are pulled at a time. You can modify the advanced parameter <code>system.process.batch_size</code> to adjust the memory usage threshold. () Important The maximum value allowed for the advanced parameter <code>system.process.batch_size</code> is 1000. You can change the value to a positive integer that is less than or equal to 1,000. The default value of system.process.batch_size is 1000.

CPU utilization	The CPU utilization threshold for a concurrent unit of a data transformation job is 100%. If you have higher requirements for CPU utilization, you can increase the number of data transformation jobs that can concurrently run based on the preceding descriptions.
Data amount in a dimension table	The maximum number of data entries allowed in a dimension table is one million, and the maximum memory that can be occupied by data in a dimension table is 1 GB. If the number of data entries exceeds one million or the memory occupied by data exceeds 1 GB, truncation is performed. In this scenario, only the one million data entries in a dimension table or the data that occupies the 1 GB of memory for a dimension table can be used. The functions that are involved include res_rds_mysql, res_log_logstore_pull, and res_oss_file. For more information, see res_rds_mysql, res_log_logstore_pull, and res_oss_file.
Result data writing	
Item	Description
Item Data writing to a destination Logstore	Description When transformation results are written to a destination Logstore, the write limits of the Logstore cannot be exceeded. For more information, see Data read and write. If you configure the hash_key_field or hash_key parameter and specify the KeyHash mode when you call the e_output and e_coutput function to write data to a destination Logstore, we recommend that you appropriately allocate hash keys and shards to minimize unbalanced data distribution.

4.5.5. Configure preview modes

You can use the preview feature to debug data transformation scripts. This feature supports the Advanced preview mode.

Advanced preview mode

In Advanced preview mode, Log Service accesses the specified Logstore and reads data from the Logstore to test the data and simulate the data

- transformation process.
- 1. Log on to the Log Service console.
- 2. Go to the data transformation page.
 - i. In the Projects section, click the project that you want to manage.
- ii. Choose Log Storage > Logstores. On the Logstores tab, click the Logstore that you want to manage.
- iii. On the Search & Analysis page, click Data Transformation.
- 3. In the upper-right corner of the page, specify a time range for the log data that you want to transform.
- Make sure that log data exists on the **Raw Logs** tab. 4. In the code editor, enter a data transformation statement.
- For more information, see Data transformation syntax.
- 5. Preview data.
 - i. In the upper-right corner of the page, click Advanced.
- ii. Click Preview Data
- iii. In the Add Preview Settings panel, configure the parameters and click OK. The following table describes the parameters.

The first time you preview data, you must configure the parameters. After you configure the settings, you can click **Modify Preview Settings** to modify the parameters.

Parameter	Description
Advanced Parameter Settings	Log Service allows you to set the passwords that are required in the transformation statement in the key-value pair format. For example, you can set a password that is used to connect to a database in the key-value pair format. You can reference passwords in the transformation statement by using the $s\{key\}$ variable. You can click the plus sign (+) to add more key-value pairs. For example, you can add config.vpc.vpc_id.test1:vpc-uf6mskb0b****n9yj, which indicates the ID of the virtual private cloud (VPC) to which an ApsaraDB RDS instance belongs.

- After you configure the preview settings, you can preview the data transformation results on the Transformation Results tab.
- If the data fails to be transformed because the syntax of the transformation statement or the permissions are invalid, troubleshoot the failure as
 prompted.
- If the data is transformed as expected, you can save the transformation statement as a rule. For more information, see Create a data transformation job.

4.5.6. Create a data transformation job

Log Service allows you to create a data transformation job to read data from a source Logstore and write transformed data to one or more destination Logstores. You can also query and analyze the transformed data to create more value. This topic describes how to create a data transformation job in the Log Service console.

Prerequisites

Data is collected to Log Service. For more information, see Data collection.

Procedure

1. Log on to the Log Service console

2. Go to the data transformation page.

- i. In the Projects section, click the project that you want to manage.
- ii. Choose Log Storage > Logstores. On the Logstores tab, click the Logstore that you want to manage.
- iii. On the query and analysis page, click **Data Transformation**.

3. In the upper-right corner of the page, specify a time range for the log data that you want to transform.

Make sure that log data exists on the Raw Logs tab.

- 4. In the code editor, enter a data transformation statement.
 - For more information, see Data transformation syntax.
- 5. Click Preview Data.
 - Log Service supports the Advanced preview mode. For more information, see Advanced preview mode.
 - Preview the transformation results.
 - If the data fails to be transformed because the syntax of the transformation statement or the permissions are invalid, troubleshoot the failure as prompted.
 - $\circ~$ If the transformed data is returned as expected, go to Step 6.
- 6. Create a data transformation job.
 - i. Click Save as Transformation Job.
 - ii. In the Create Data Transformation Job panel, configure the parameters and click OK. The following table describes the parameters.

Parameter	Description	
Job Name	The name of the data transformation job.	
Storage Target		
Target Name	 The name of the storage destination. Storage Target includes Target Project and Target Logstore. You can create multiple storage destinations to store the transformed data in different destination Logstores. You can also use the name parameter of the e_output or e_coutput function in the transformation statement to specify the name of the storage destination. For more information, see e_output and e_coutput. If you do not include the e_output function in the transformation statement, the job writes the transformed data to the Logstore in the storage destination Logstore, you do not need to include the e_output function in the transformation statement. If you want to configure only one destination Logstore, you do not need to include the e_output function in the transformation statement. If you include the e_output or e_coutput function and configure thename, project, and logstore parameters for the function, the job runs based on the parameter settings in the functions even if you configure the Target Project and Target Logstore parameters in this step. 	
Target Region	The region where the destination project resides. If you want to perform data transformation across regions, we recommend that you use HTTPS for data transmission. This ensures the privacy of log data.	
Target Project	The name of the destination project to which transformed data is saved.	
Target Logstore	The name of the destination Logstore to which transformed data is saved.	
Processing Range		
Time Range	 The time range within which the data is transformed. Valid values: Note The value of Time Range depends on the time when logs are received. All: transforms data in the source Logstore from the first log until the job is manually stopped. From Specific Time: transforms data in the source Logstore from the log that is received at the specified start time until the job is manually stopped. Within Specific Period: transforms data in the source Logstore from the log that is received at the specified start time to the log that is received at the specified start time to the log that is received at the specified start time to the log that is received at the specified end time.	
Advanced		
Advanced Parameter Settings	Log Service allows you to set the passwords that are required in the transformation statement in the key-value pair format. For example, you can set a password that is used to connect to a database in the key-value pair format. You can reference passwords in the transformation statement by using the $\{key\}$ variable. You can click the plus sign (+) to add more key-value pairs. For example, you can add config.vpc.vpc_id.test1:vpc-ut6mskb0b****n9yj, which indicates the ID of the virtual private cloud (VPC) to which an ApsaraDB RDS instance belongs. Advanced config.vpc.vpc_id.test1 23/100 : vpc-ut6mskb0b****n9yj 21/2000 X Parameter Settings	

What to do next

After you create a data transformation job, you can perform the following operations:

- On the **Data Transformation Overview** page, view the details and status of the job. You can also modify or stop the job. For more information, see Manage a data transformation job.
- In a destination Logstore, perform query and analysis operations. For more information, see Query and analyze logs.

4.5.7. Manage a data transformation job

This topic describes how to manage a data transformation job in the Log Service console. You can view the details and status of a job in the console. You can also start, stop, modify, and delete the job and configure alert rules.

View the details of a data transformation job

- 1. Log on to the Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. In the left-side navigation pane, choose **Jobs > Data Transformation**.
- 4. In the data transformation job list, find and click the data transformation job.
- 5. On the Data Transformation Overview page, view the details of the data transformation job.

Parameter	Description
Source Logstore	The name of the Logstore from which data is read for transformation.
Status	The status of the data transformation job. For more information, seeView the status of a data transformation job.
Job ID	The ID of the data transformation job.
Time Range	The time range within which the data transformation job is run.
Job Created At	The time when the data transformation job was created.
Job Modified At	The time when the data transformation job was last modified.
Job Ended At	The time when the data transformation job ends. Note If you do not configure the End Time parameter when you create a data transformation job, the value of this parameter is empty.
Storage Target	The name of the storage destination, the name of the Logstore where transformed data is stored, and the name of the project to which the Logstore belongs.
Consumption Progress	The data consumption progress of the data transformation job.
Status	The dashboard that is provided by Log Service to display the statistics about the execution of the data transformation job. The statistics include the overall information, transformation rate, consumption latency, consumption rate, active shards, and exceptions.

View the status of a data transformation job

You can view the status of a data transformation job on the **Data Transformation Overview** page. The following table lists the job states and the operations supported in each state.

State/Action	Stop the job	Start the job	Rerun the job	Modify the job	Delete the job
Starting	Not supported	Not supported	Not supported	Supported	Supported
Running	Supported	Not supported	Supported	Supported	Supported
Stopping	Not supported	Not supported	Not supported	Supported	Supported
Terminated	Not supported	Supported	Supported	Supported	Supported
Success	Not supported	Not supported	Supported	Supported	Supported
Failed	Not supported	Not supported	Supported	Supported	Supported

Stop a data transformation job

If a data transformation job is in the Running state, click Stop on the Data Transformation Overview page of the job.

? Note

If a data transformation job is stopped, the system records the checkpoint at which the job is stopped. If the job is started again, data transformation is resumed from this checkpoint. If you want to resume data transformation from the beginning, you must rerun the job. For more information, see Rerun the data transformation job.

Start a data transformation job

If a data transformation job is in the Terminated state, click Start on the Data Transformation Overview page of the job.

Rerun the data transformation job

You can rerun a data transformation job regardless of the job status. To rerun a data transformation job, choose **More > Rerun** on the **Data Transformation Overview** page of the job.

? Note

If you rerun a data transformation job, the job is run from the beginning. If you want to continue with a data transformation job from the checkpoint at which the job was stopped, you must stop the job and then start the job.

Modify the transformation rule of a data transformation job

? Note

After you modify the transformation rule of a data transformation job, you can perform the following operations:

- If you want to use the new transformation rule to continue with the job, you must stop the job and then start the job.
- If you want to use the new transformation rule to rerun the job, you must run the job from the beginning. For more information, seeRerun the data transformation job.

1. On the Data Transformation Overview page, click Edit Rule.

2. Modify the transformation rule based on your business requirements.

For more information about the transformation rule syntax, see Data transformation syntax.

- Preview data.
 - i. Select Quick.

Log Service supports the Advanced preview mode. For more information, see Advanced preview mode.

- ii. Click Preview Data.
- iii. Preview the transformation results.
 - If the data fails to be transformed because the syntax of the transformation rule or the permissions are invalid, troubleshoot the failure as prompted.
- If the transformed data is returned as expected, go to Step 5.

4. Click Modify Transformation Settings.

5. Modify the settings and click OK.

For more information about the parameters, see Create a data transformation job.

Delete a data transformation job

```
Marning
```

After you delete a data transformation job, the tjob cannot be restored. Proceed with caution.

You can delete a data transformation job by using one of the following methods:

• In the data transformation job list, move the pointer over the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to delete and choose Note that the data transformation job that you want to de

• On the Data Transformation Overview page, choose More > Delete.

4.5.8. Data transformation syntax

4.5.8.1. Language overview

Domain-specific language (DSL) for Log Service is a Python-compatible scripting language that is used for data transformation in Log Service. The Python-based domain-specific language (DSL) for Log Service provides more than 200 built-in functions that you can use to efficiently transform data.

Flexible orchestration

You can use DSL for Log Service to edit functions in a flexible manner and combine functions to implement complex logic in most data transformation scenarios.

Dynamic distribution

You can use DSL for Log Service to distribute data to different Logstores based on specific logic and your business requirements. The names of the Logstores can be obtained by using dynamic computing or from external resources such as Object Storage Service (OSS) buckets.

Data enrichment

- You can use DSL for Log Service to obtain data for enrichment from local or external resources, such as OSS buckets and ApsaraDB RDS for MySQL instances.
- You can use DSL for Log Service to perform regular mapping for dictionaries and tables and advanced mapping for tables.
- You can use DSL for Log Service to automatically refresh external resources that are loaded.

Global processing functions

DSL for Log Service provides approximately 30 global processing functions. You can configure the parameters of global processing functions to control processing operations. Global processing functions accept the results of expression functions as parameters. Control functions are a type of global processing functions and can be used together with expression functions and the following types of global processing functions:

Flow control functions

- You can control processes based on conditions by using functions such as <code>e_if_else</code> , <code>e_if</code> , <code>e_switch</code> , and <code>e_compose</code> .
- You can use simple search functions, such as e_search , to process different types of logs in a flexible manner.
- Event processing functions

You can discard, retain, split, write, and replicate events.

Field processing functions

You can retain, delete, and rename fields.

- Value extraction functions
- You can extract values or key-value pairs from fields based on regular expressions, Grok patterns, syslog protocols, quotes, key-value pair delimiters, and delimiters such as commas (,), vertical bars (]), and tabs (\t).
- You can extract and enrich JSON data.
- Mapping and enrichment functions
 - $\circ~$ You can map or search for data based on a dictionary or a table.
- You can obtain the information about a dimension table that is used to enrich data from resources such as rule configurations, external OSS buckets, and ApsaraDB RDS for MySQL instances.
- You can use a function to automatically refresh external resources based on full or incremental change logs.
- Value-added content function

You can enrich the information about some log fields. For example, you can obtain threat intelligence for an IP address and store the threat intelligence to log fields for log analysis.

Expression functions

DSL for Log Service provides more than 200 built-in expression functions to convert events or affect the results of the global processing functions. The expression functions are suitable for most data transformation scenarios. DSL for Log Service provides the following expression functions:

Event check functions

- DSL for Log Service provides a condition-based filtering mechanism that uses Lucene-like syntax, complete regular expressions, strings, generic characters, numeric value comparison, and logical operators such as AND, OR, and NOT.
- Operator functions

You can extract, control, and compare field values. You can also perform container evaluation and operations on multiple fields.

Conversion functions

You can convert the values of basic data types. You can also convert numbers, dictionaries, and lists.

• Arithmetic functions

You can perform basic, multi-value, and mathematical calculations. You can also perform operations based on mathematical parameters.

String functions

You can encode, decode, sort, reverse, replace, normalize, search, evaluate, truncate, and format multiple fields. You can also perform evaluation based on character sets.

Date and time functions

You can convert date and time values. You can obtain date and time attributes, date and time values, UNIX timestamps, and date and time strings. You can also modify and compare date and time values.

Regular expression functions

You can extract, match, evaluate, replace, and truncate fields.

Grok function

DSL for Log Service provides more than 400 built-in Grok patterns. Grok patterns can be replaced.

Structured data functions

You can extract and filter JSON, Protobuf, and XML data.

- IP address parsing functions
- You can parse IP addresses and convert data.
- Encoding and decoding functions

You can encode and decode text in the SHA1, SHA256, SHA512, MD5, HTML, URL, or Base64 format.

4.5.8.2. Syntax overview

The Python-based domain-specific language (DSL) for Log Service provides more than 200 built-in functions that you can use to efficiently transform data. This topic describes the language modes, function categories, and implementation of the DSL for Log Service.

Language modes

The DSL for Log Service is compatible with Python. In standard mode, the DSL can be regarded as a subset of Python. Except for basic data structures and expressions, other syntax rules are orchestrated by using functions.

Category	Python syntax	Standard mode
Data structure	Number, string, and Boolean	Supported. Strings that start or end with """ are not supported.
	Tuple, list, set, and dictionary	Supported. The set structure is not supported. Example: {1,2,3} .
	Object	Only built-in extended data structures such as table and datetime objects are supported.
	Operators such as the plus sign (+), the subtraction sign (-), the multiplication sign (\times), and the divide operator (/)	Only comparison operators such as ==, !=, and > , and logical operators such as AND, OR, and NOT can be directly used in code. You must call functions to use the functionality of other operators.
	Comments	Supported.

User Guide-Log Service

	Variable assignment	Not supported. You must call functions to assign values to variables.
Basic syntax	Condition evaluation	Supported. Functions: e_{if} , e_{if} _else, and e_{switch} .
		Indirectly supported. You must use nested built-in functions to implement loops. The following sample functions show how to traverse the elements in an array:
	Loops	<pre>e_if(op_ge(op_len(json_parse(v("x"))), 1), e_set("x0", lst_get(v("x"), 0))) e_if(op_ge(op_len(json_parse(v("x"))), 2), e_set("x1", lst_get(v("x"), 1)))</pre>
Function	Standard built-in functions of Python	Not supported. You can use more than 200 built-in functions provided by the DSL for Log Service.
	Function calls	Supported. Function calls that use parameter unpacking are not supported.
	UDFs, such as def or lambda	Not supported. You can use more than 200 global processing and expression functions provided by the DSL for Log Service. You can also combine these functions based on your business requirements.
	Import and use of the Python standard library	Not supported.
Module	Creation of threads and processes	Not supported.
	Import of third-party libraries	Not supported.
	External network connection or external command call	Supported. The DSL for Log Service provides built-in resource connectors.

Function categories

In standard mode of the DSL for Log Service, all operations are performed by calling functions. The DSL provides more than 200 built-in functions, which are categorized into global processing functions and expression functions.

Global processing functions

Global processing functions are used to receive, process, and return logs. Only global processing functions can be used to construct each step in a transformation rule.

• Expression functions

Expression functions are commonly used to receive specific parameters and return specific values. Expression functions can be combined and passed to global processing functions as parameters to define more flexible logic.

The following table describes global processing functions and expression functions.

Category	Construct a step	Receive a log	Return results	Modify a log	Combine functions
Global processing functions	Supported.	Logs are automatically received.	Zero to multiple logs are returned.	Supported. In most cases, logs can be modified.	Supported.
Expression functions	Not supported.	Supported by only a few expression functions. Most expression functions do not directly process logs.	Specific data structures are returned.	Not supported.	Supported.

Global processing functions

Global processing functions are used to receive, process, and return logs.

(?) Note Only a global processing function can be placed in the first line of each step.

The following syntax is used:

```
Global Processing Function 1(..Parameters....)
Global Processing Function 2(..Parameters....)
Global Processing Function 3(..Parameters....)
Global Processing Function 4(..Parameters....)
```

Global processing functions can be further categorized into flow control functions and event processing functions. The following table describes the functions.

Category	Description	Example
Flow control functions	The functions are used to manage processes, receive logs, and call other functions to process logs based on specific conditions.	<code>e_if</code> , <code>e_switch</code> , and <code>e_if_else</code> .
Event processing functions	The functions are used to transform logs. Zero to multiple logs are returned.	<pre>Examples: e_drop_fields : discards log fields. e_kv : extracts the key-value pairs of logs. e_dict_map : enriches logs.</pre>

Transformation logic:

• Basic processing The data transformation feature reads streaming data from a source Logstore and sends each log in a dictionary structure to specific functions. Then, the feature runs the functions that are specified in the transformation rule in sequence to process the events and writes the transformation results to specified destination Logstores.

(2) Note All fields and values of a log are sent as strings. For example, the raw log {"__time__": "1234567", "__topic__": "", "k1": "test"} is processed by the e_set("f1", 200) function. The function adds the f1 field whose value is set to 200 to the raw log. Then, the raw log is transformed into {"__time__": "1234567", "__topic__": "", "k1": "test", "f1": "200"}. In this log, the f1 field and the value 200 are strings.

The event processing functions specified in a transformation rule are called in sequence. Each function receives and processes a log, and then returns a processed log.

For example, the $e_{set}("type", "test")$ function adds the type field whose value is set to test to a log. The next function receives and processes the processed log.

- Condition evaluation
- e_if: You can call the e_if function to add conditional expressions to process logs. If a log does not meet the specified condition, the corresponding operation is skipped. The e_if function implements the if logic.

For example, the $e_{if(e_{match}("status", "200"), e_{regex}("data", "ret: \d+", "result"))}$ function checks whether the value of the status field is 200. If the value is 200, the function extracts the result field from the data field by using the specified regular expression. Otherwise, no operation is performed.

- $\circ \quad {\tt e_if_else} \ : {\tt This \ function \ works \ in \ a \ manner \ similar \ to \ the} \quad {\tt if_else} \quad {\tt function.}$
- Processing termination
- $\circ\,$ A step in a transformation rule may return no log. This indicates that the related log is deleted.

For example, the <code>e_if(str_islower(v("result")), e_drop())</code> function is used to check whether the value of the <code>result</code> field in a log is a string that consists of only lowercase characters. If the condition is evaluated to true, the log is discarded and the subsequent steps are not performed on this log. The system automatically processes the next log.

If a log is written to a destination Logstore, the processing is terminated. For example, if the e_output function is used to write a log to a destination Logstore and delete the log, the subsequent steps are not performed on this log.

③ Note The e_coutput function copies the output log and the subsequent steps are performed on this log.

Log splitting for parallel processing

A step in a transformation rule may return multiple logs. This indicates that the related log is split.

For example, the $e_{\texttt{split}(data)}$ function splits a log into two logs based on the value of the data field. If the value of the data field in the log is "abc, xyz", the log is split into two logs. In one log, the value of the data field is abc. In the other log, the value of the data field is xyz.

The logs that are generated after splitting are processed in the subsequent steps.

Expression functions

In addition to global processing functions, the DSL for Log Service provides 200 expression functions that are used to receive specific parameters and return specific values. You can call an expression function or a combination of expression functions in a global processing function. The following syntax is used:

Global Processing Function 1 (Expression Function 1 (...), ...) Global Processing Function 2 (..., Expression Function 2 (...), Expression Function 3 (...), ...)

Expression functions can be categorized into event check functions, resource functions, control functions, and other expression functions. The following table describes the functions.

Category	Description	Example
Event check functions	The functions are used to receive logs, extract specific information, and then return the information without modifying the logs.	v : returns the value of a log field. e_match : checks whether the value of a field in a log meets a specified condition.
Resource functions	The functions are used to access on-premises or external resources, receive specific parameters, and return specific values. The data types of the return values include dictionary and table.	res_oss_file, res_rds_mysql, and res_log_logstore_pull.
Control functions	The functions are used to receive specific parameters and perform logical operations on expressions or condition-based control. The functions are also used to call other expression functions to return results.	op_and , op_or , op_not , op_if , and op_coalesce .
Other expression functions	The functions are used to receive specific parameters or the results of other functions and return specific values.	String functions, date and time functions, and conversion functions.

4.5.8.3. Data structures

This topic describes the data structures that are related to the data transformation syntax.

Basic data structures

The following table describes the different types of basic data structures.

Туре	Description
Integer	You can use integers as field values. You can also pass integers as values of parameters to functions. For example, e_set("f1", 100) indicates that the f1 field is set to100.
Float	You can use float values as field values. You can also pass float values as values of parameters to functions. For example, $e_set("fl", 1.5)$ indicates that the fl field is set to 1.5.

String	<pre>Strings can be specified in multiple formats. Examples: "abc" is equivalent to 'abc' . If a string contains a double quotation mark ("), you can specify the string in the 'abc"xyz' format. You can also use a backslash (\) to escape the double quotation mark in the "abc\"xyz" format. Backslashes (\) are used to escape special characters. For example, "\\abc\\xyz" indicates the \\abc\xyz string. r"\\10.64.1.1\share\folder" and "\\\10.64.1.1\\share\\folder" indicate the \\10.64.1.1\share\folder string. Multibyte strings are encoded in Unicode. For example, the length of a string that consists of two chinese characters is 2. Regular expressions are represented as strings. for Note A field value must be enclosed in double quotation marks (""). You can use single quotation marks(") to enclose the field value. For example, e_search ("domain: '/url/test.jsp'') is invalid, and e_search('domain: "/url/test.jsp"') </pre>
Byte	Example: <code>b'abc'</code> . Bytes are encoded in memory by using a format that is different from the format of strings. Bytes are received and returned by special functions.
None	None and null indicate a null value. Some named parameters of functions use None as the default value to indicate a specific default behavior. ⑦ Note None or null is different from an empty string.
List	 An array. Example: [1,2,3,4]. Some functions accept lists as parameters. Example: e_dict_map("dict data", ["f1", "f2", "f3"],) Some functions return lists. For example, if you call the json_select function to extract an array, a list is returned.
Tuple	Tuples and lists function in the same manner. Example: (1, 2, 3, 4) .
Dictionary	 A dictionary is a collection of key-value pairs in the {"key": "value", "k2": "v2",} format. Keys are strings in most cases and cannot be repeated. The values of keys can be of the preceding data types. The key-value pairs are stored in a hash table in an unordered manner. An event is a special dictionary. Some functions accept dictionaries in specific formats. Example: {"key": [1,2,3], "ke": {"k3": "va3"} }. The dictionary structure is used as the input data to map fields to a dictionary.
Boolean	Examples: True , False , true , and false .
Table	Each table consists of multiple columns. You can construct a table by loading multi-row CSV-formatted data from an external resource. You can also construct a table by loading multiple columns of data from ApsaraDB RDS instances and Logstores. Tables are suitable for advanced operations such as data mapping and enrichment.
Datetime object	A datetime object is a memory object that indicates date and time information. A datetime object can be converted to a UNIX timestamp or a formatted time string. A datetime object can be passed to dt_{-} -like functions for further conversion.

Event types

The following list describes event types:

Basic types

Log data is processed into the dictionary structure during the data transformation process. Example: {"_topic_": "access_log", "content": "...."} .

The keys and values of a dictionary correspond to the fields and values in a log.

⑦ Note The keys and values of an event are strings, and the keys must be unique.

- Meta-fields
 - The following meta-fields are supported:
 - __time__ : the log time that is specified when log data is written. The value is a string that represents an integer and follows the UNIX time 0 format. It is the number of seconds that have elapsed since 00:00:00 UTC, Thursday, January 1, 1970.
 - : the topic of a log. Topics are used to group logs in a Logstore. You can specify a topic for logs when the logs are written to a • topic Logstore. You can specify a topic when you query logs.
- Modification of the __time__ field You can change the value of the __time__ field to modify the event time of a log. You can use date and time functions to perform more operations on the __time__ field.

③ Note If the _time_ field is deleted, the system time at which a log is processed is used as the event time at which the log is written to a destination Logstore.

Tags

Tags are used to differentiate fields in logs. Tags are in the __tag_:Name format.

- If the source Logstore is configured to record public IP addresses, logs contain tag: _tag_:_receive_time_ .
- Container logs contain a large number of container-related tags. Example: tag : container name .

 $\circ \ \mbox{You can add and modify tags. For example, you can add a tag named type: \ \mbox{e_set("_tag_:type", "access_log")} \ .$

• Automatic conversion during value assignment The keys and values of an event are strings. When you assign a value to a key or when you specify a new value for a key in an event, the key and the value of the key are automatically converted to strings. Examples:

e_set("v1", 12.3) e_set("v2", True)

Set v1 to the 12.3 string, and set v2 to the true string.

The following table provides examples on the conversion of different data types to strings.

Original type	Example	New type	Example
Integer	1	String	"1"
Float	1.2	String	"1.2"
Boolean	True	String	"true"
Byte	b"123"	String that is encoded in UTF-8	"123"
Tuple	• Example 1: (1, 2, 3) • Example 2: ("a", 1)	String that represents a list	 Example 1: "[1, 2, 3]" Example 2: "[\"a\", 1]"
List	• Example 1: [1,2,3] • Example 2: ["a", 1]	String	 • Example 1: "[1, 2, 3]" • Example 2: "[\"a\", 1]"
Dictionary	{"1":2, "3":4}	String	"{\"1\": 2, \"3\": 4}"
Date and time	datetime(2018, 10, 10, 10, 10, 10, 10)	String that represents time in the ISO format	2018-10-10 10:10:10

Fixed identifiers

The data transformation feature provides fixed identifiers. You can use the identifiers to simplify code.

Identifier	Туре	Description
true	Boolean	Equivalent to True .
false	Boolean	Equivalent to False .
null	None	Equivalent to None .
F_TAGS	String	The regular expression that represents the ${\tt TAG}$ field, equivalent to "tag_:.+" .
F_META	String	The regular expression that represents the combination of the TAG ,topic , andsource fields, equivalent totag_:.+!_topic!_source
F_TIME	String	The name of thetime field, equivalent totime
F_PACK_META	String	The regular expression that represents the pack meta field, equivalent to "_pack_meta_!_tag_:_pack_id_" .
F_RECEIVE_TIME	String	The tag field of the time at which a server receives a log, equivalent to "_tag_:_receive_time_" .
C_JOB_REGION	String	The region ID of a data transformation job. Example: cn-hangzhou . For example, e_set ("job_region", C_JOB_REGION) assigns the region ID of a data transformation job to the job_region field.
C_JOB_PROJECT	String	The name of the project to which a data transformation job belongs. Example: my-sls- project .For example, e_set("job_project", C_JOB_PROJECT) assigns the name of the project to which a data transformation job belongs to the job_project field.
C_JOB_NAME	String	The configuration name of a data transformation job. Example: etl-1649227848-642277 . For example, e_set("job_name", C_JOB_NAME) assigns the configuration name of a data transformation job to the job_name field.
C_JOB_ID	String	The running ID of a data transformation job. Example: 73b96061b8c1c2101d558139bf641ea9 . For example, e_set("job_id", C_JOB_ID) assigns the running ID of a data transformation job to the job_id field.

JSON objects

A JSON object is an object that you obtain after the JSON expression function <code>json_select</code> or <code>json_parse</code> is used to parse data. A JSON object consists of data in basic data structures. The following table provides examples on the conversion of strings to JSON objects.

String	JSON object	Туре
1	1	Integer
1.2	1.2	Float
true	True	Boolean
false	False	Boolean
"abc"	"abc"	String
null	None	None
["v1", "v2", "v3"]	["v1", "v2", "v3"]	List
["v1", 3, 4.0]	["v1", 3, 4.0]	List
{"v1": 100, "v2": "good"}	{"v1": 100, "v2": "good"}	Dictionary
{"v1": {"v11": 100, "v2": 200}, "v3": "good"}	{"v1": {"v1": 100, "v2": 200}, "v3": "good"}	Dictionary

4.5.8.4. Basic syntax

This topic describes the basic syntax of the domain-specific language (DSL) for Log Service.

Comments

Start the comment of a step with a number sign (#). Examples:

```
# Specify the default topic. This is a comment at the beginning of a step.
```

e_set("_topic_", "access_log") # Specify the default topic. This is a comment at the end of a step.

Line wrapping

- If the parameter list or string of a function does not fit on a single line, you can separate the parameter list or the string.
- If the parameter list contains a comma (,), you can split the parameter list immediately after the comma (,).

• If you want to split a string, use a backslash (\) to indicate that the string continues in the next line.

```
Examples:
```

Function invoking

Invoke the basic functions

```
e_set("abc", "xyz")
```

③ Note When you write a data transformation statement, the data types and the number of parameters that you pass to a function must meet the syntax of the function.

· Pass the basic variable parameters

str_replace(value, old [,new [,count]])

(2) Note The parameters that are enclosed in the square brackets [] are optional. For example, the new and count parameters in the preceding function are optional. You cannot pass these parameters the same way you pass the named parameters. You must pass these parameters in sequence.

```
# Invalid examples
str_replace("a-b-c", "-", new='%')
str_replace("a-b-c", "-", new='%', count=1)
# Valid examples
str_replace("a-b-c", "-", '%')
str_replace("a-b-c", "-", '%', 2)
```

· Pass the named parameters

A parameter that has a default value is called a named parameter. For example, the mode parameter in the e_set("abc", "xyz", mode="fill") function is a named parameter.

 You must pass the values of the named parameters in specific functions based on specific conditions. For more information, see the parameter description of each function.

- $\circ\,$ You can pass the value of a named parameter when you configure a parameter in the format of $_{\tt mode=...}$
- You can pass multiple named parameters in random order. For example, e_csv("data", ["f1", "f2", "f3"], sep='#', quote="|") is equivalent to e_csv("data", ["f1", "f2", "f3"], quote="|").

(?) Note The named parameters follow the non-named parameters.

Invoke a combination of functions

You can pass the returned value of a function as the value of a parameter to another function. In this case, you must make sure that the returned value is of the same data type as the value of the parameter. Examples:

e_set("abc", v("xyz"))

e_set("abc", str_lower(v("xyz")))

Variable parameters

You can pass variable parameters to specific functions. The v("fl", ...) function specifies that multiple parameters can be passed. Example: v("f1", "f2", "f3")

If you need to pass both variable parameters and named parameters, you must place the named parameters after the variable parameters. Example: v("f1", "f2", "f3", "f4", mode="fill") .

Operators

· Comparison operators

The following comparison operators are supported in the DSL for Log Service in standard mode: >, <,>=, <=, ! =,== . You can also use the comparison functions that are provided by Log Service to perform the operations.

Use comparison operators

The following examples show how to use comparison operators. If the comparison condition is evaluated to True, the related log is discarded.

e_if(3 > 2, DROP)	# If 3 is greater than 2, the log is discarded.
e_if(3 < 2, DROP)	# If 3 is less than 2, the log is discarded.
e_if(3 >= 2, DROP)	# If 3 is greater than or equal to 2, the log is discarded.
e_if(3 <= 2, DROP)	# If 3 is less than or equal to 2, the log is discarded.
e_if(3 == 2, DROP)	# If 3 is equal to 2, the log is discarded.
e_if(3 != 2, DROP)	# If 3 is not equal to 2, the log is discarded.
e_if(1 < 2 < 3, DROP)	# If 2 is greater than 1 and 2 is less than 3, the log is discarded.
if(0 < ct int(v(!x!)))) < 100 DDOD) # If the value of the v field is greater than 0 and less than 100 the log is discarded

· Use the comparison functions that are provided by Log Service

Operation	Function	Example
Equal to (==)	op_eq	<pre>op_eq(v("name"), "xiao ming")</pre>
Not equal to (!	op_ne	op_ne(v("name"), "xiao ming")
Greater than (>)	op_gt	<pre>op_gt(ct_int(v("age")),)</pre>
Greater than or equal to (>=)	op_ge	<pre>op_ge(ct_int(v("age")), 18)</pre>
Less than (<)	op_lt	<pre>op_lt(ct_int(v("age")), 18)</pre>
Less than or equal to ($\langle <= \rangle$)	op_le	<pre>op_le(ct_int(v("age")), 18)</pre>

Logical operators

The following logical operators are supported in the DSL for Log Service in standard mode: AND, OR, and NOT. You can also use the logical functions that are provided by Log Service to perform the operations.

Use logical operators

The following examples show how to use logical operators. If the logical condition is evaluated to True, the related log is discarded.

e_if(True or False, DROP) # True is returned.

e_if(True and not False, DROP) # True is returned.

 $e_{if}(3 > 2 \text{ and } 1 < 3, \text{ DROP})$ # True is returned.

 $e_{if}(ct_{int}(v('x')) > 100 \text{ or } ct_{int}(v('y')) < 100, DROP) # If the value of the x field is greater than 100 or the value of the y field is$ less than 100, True is returned.

Use the logical functions that are provided by Log Service

Operation	Function	Example
Logical operator AND (and)	op_and	<pre>op_and(op_gt(v("age"), 18), op_lt(v("age"), 31))</pre>
Logical operator OR (or)	op_or	<pre>op_or(op_le(v("age"), 18), op_gt(v("age"), 65))</pre>
Logical operator NOT (not)	op_not	<pre>op_not(op_gt(v("age"), 18))</pre>

Other operators

You cannot directly use specific DSL operators in standard mode. Log Service provides functions that you can use to perform the operations. The following table describes the operators and functions

Operation	Function	Example
Addition (+)	op_add	op_add(v("age"), 2)
Subtraction (-)	op_sub	op_sub(v("age"), 2)
Multiplication (*)	op_mul	<pre>op_mul(v("size"), 2)</pre>

User Guide-Log Service

Cloud Defined Storage

Exponentiation (**)	op_pow	op_pow(v("size"), 2)
Floor division (//)	op_div_floor	<pre>op_div_floor(v("bytes"), 1024)</pre>
Modulus (%)	op_mod	op_mod(v("age"), 10)
Negation (-)	op_neg	<pre>op_neg(v("profit"))</pre>
Existence check (in)	op_in	<pre>op_in(["pass", "ok"], v("result"))</pre>
Nonexistence check (not in)	op_not_in	<pre>op_not_in(["pass", "ok"], v("result"))</pre>
String slicing ([])	op_slice	op_slice(v("message"), 0, 20)

In this example, the value of the a field is 3600 * 6. The following examples show an invalid function and a valid function to specify the value for the field.

# *	
e_set("a", 3600 * 6)	
e_set("a", op_mul(3600, 6)) # Valid	
# /	
e_set("bytes_kb", v("bytes") / 1024)	# Invalid
<pre>e_set("bytes_kb", op_div_floor(v("bytes"), 1024))</pre>	# Valid

True or false evaluation

Some functions check whether a condition is true or false to specify the event processing logic. A condition can be a fixed value or a value returned by an expression function.

You can perform true or false evaluation for all types of data in the DSL for Log Service orchestration. The following table describes the rules for true or false evaluation.

Data type	True	False
Boolean	True, true	False, false
None	None	Always false
Numeric	Not 0 or 0.0	0 or 0.0
String	Not empty	Empty string
Bytes	Not empty	Empty bytes
Tuple	Not empty	Empty tuple
List	Not empty	Empty list
Dictionary	Not empty	Empty dictionary
Table	One or more tables exist	No table exists
Datetime	One or more datetime objects exist	No datetime object exists

The following functions can be used to discard logs based on the conditions:

e_if(v("abc"), DROP)

e_if(True, DROP) e if(1, DROP)

If the value of the first parameter is True, the log is discarded.
If the value of the first parameter is 1, the log is discarded.
#If the abc field exists and the value of this field is not empty, the log is discarded.

e_if(str_isdigit(v("abc")), DROP) # If the abc field exists and the value of this field contains only digits, the log is discarded.

4.5.8.5. Function overview

This topic describes the functions that you can call to transform data in Log Service.

Global processing functions

Category	Function	Description
	e_if	 Performs an operation if a specified condition is met. You can specify multiple condition-operation pairs. If a condition is met, the function performs the operation that corresponds to the condition. If the condition is not met, the function does not perform the operation, but evaluates the next condition. If the function performs an operation that deletes a log, the function no longer performs other operations on the log.
	e_if_else	Performs an operation based on the evaluation result of a condition.
Flow control functions	e_switch	 Performs an operation if a specified condition is met. You can specify multiple condition-operation pairs. If a condition is met, the function performs the operation that corresponds to the condition and returns the result. If the condition is not met, the function does not perform the operation, but evaluates the next condition. If no specified conditions are met and the default parameter is configured, the function performs the operation that is specified by the default parameter and returns the result. If the function performs an operation that deletes a log, the function no longer performs other operations on the log.

	e_compose	 Combines multiple operations. The function is commonly used in the e_if, e_switch, or e_if_else function. The function performs specified operations on a log in sequence and returns the result. If the function performs an operation that deletes a log, the function no longer performs other operations on the log.
	e_drop	Discards a log if a specified condition is met.
	e_keep	Retains a log if a specified condition is met.
	e_split	Splits a log into multiple logs based on the value of a specified field. You can also use the JMESPath expression to extract the value of the field, and then split the log.
Event processing functions	e_output	Writes a log to a specified Logstore. You can specify the topic, source, tags and MD5 hash keys of a shard for the log. The log is deleted after it is written to the specified Logstore. The system no longer transforms the log.
	e_coutput	Writes a log to a specified Logstore. You can specify the topic, source, tags and MD5 hash keys of a shard for the log. The log is retained after it is written to the specified Logstore. The system continues to transform the retained log.
	e_to_metric	Converts logs to metrics that can be stored in a Metricstore.
	v	Extracts the value of a field from a log. If you specify the names of multiple fields for the function, the function returns the value of the first field that exists in the log.
	e_set	Adds a field or specifies a new value for an existing field.
Field processing functions	e_drop_fields	Deletes the log fields that meet a specified condition.
	e_keep_fields	Retains the log fields that meet a specified condition.
	e_pack_fields	Encapsulates log fields and assigns the log fields as a value to a new field.
	e_rename	Renames the log fields that meet a specified condition.
	e_regex	Extracts the value of a field based on a regular expression and assigns the value to other fields.
	e_json	Manages JSON objects in a specified field in a log. You can configure the parameters to expand JSON data, extract JSON data by using the JMESPath expression, or expand the extracted JSON data.
	e_kv	Extracts key-value pairs from multiple input fields by using a specified quote.
	e_kv_delimit	Extracts key-value pairs from input fields by using a specified delimiter.
Value extraction functions	e_csv	Extracts multiple fields from a specified field by using a specified delimiter and predefined field names. The default delimiter is a comma (,).
value extraction functions	e_tsv	Extracts multiple fields from a specified field by using a specified delimiter and predefined field names. The default delimiter is \t .
	e_psv	Extracts multiple fields from a specified field by using a specified delimiter and predefined field names. The default delimiter is a vertical bar ().
	e_syslogrfc	Calculates the values of the facility and severity fields and returns the value of the facilitylabel field that indicates level information. The function calculates the values based on the value of the priority field and the specified syslog protocol.
	e_anchor	Extracts strings by using the rules specified by anchor_rules.
	e_dict_map	Maps the value of an input field to a value in a specified dictionary and returns a new field.
Mapping and enrichment functions	e_table_map	Maps the value of an input field to a row in a specified table and returns a new field.
	e_search_dict_map	Searches the keywords in a specified dictionary for a raw log field, maps the field to a value in the dictionary, and returns a new field. The keywords must be query strings.
	e_search_table_map	Searches a specified column in a specified table for a raw log field, maps the field to a row in the table, and returns a new field. The values of the column must be query strings.
Value-added content function	e_threat_intelligence	Obtains the threat intelligence for an IP address or a domain name that is specified by a log field and assigns the threat intelligence as a value to a specified field.

Expression functions

Category	Function	Description		
Event check functions	e_has	Checks whether a log field exists.		
	e_not_has	Checks whether a log field does not exist.		
	e_search	Searches for a log by using a query syntax that is similar to Lucene.		
	e_match, e_match_all, and e_match_any	Check whether the value of a log field meets the conditions specified in a regular expression.		
	op_if	Returns the value of an expression based on a specified condition.		
	op_ifnull	Returns the value of the first expression whose value is not None.		
	op_coalesce	Returns the value of the first expression whose value is not None.		
	op_nullif	Returns none if the value of Expression 1 is equal to the value of Expression 2. If the values of Expression 1 and Expression 2 are different, the value of Expression 1 is returned.		
	op_and	Evaluates the specified expressions by using the logical AND operator and returns True if all specified expressions evaluate to true. The value of each expression can be of an arbitrary data type.		
----------------------	--------------	--	--	--
	op_not	Evaluates a specified expression by using the logical NOT operator and returns the inverse Boolean value of the specified expression. The value of the expression can be of an arbitrary data type.		
	op_or	Evaluates the specified expressions by using the logical OR operator, and returns True if a specified expression evaluates to true or returns False if all specified expressions evaluate to false. The value of each expression can be of an arbitrary data type.		
	op_eq	Returns True or False based on the a==b condition.		
	op_ge	Returns True or False based on the a>=b condition.		
Operator functions	op_gt	Returns True or False based on the a>b condition.		
	op_le	Returns True or False based on the a<=b condition.		
	op_lt	Returns True or False based on the a condition.		
	op_ne	Returns True or False based on the a !=b condition.		
	op_len	Calculates the number of characters in a text string. This function applies to strings or expressions that return tuples, lists, or dictionaries.		
	op_in	Checks whether a string, tuple, list, or dictionary contains a specified element and returns True or False.		
	op_not_in	Checks whether a string, tuple, list, or dictionary does not contain a specified element and returns True or False.		
	op_slice	Extracts strings from a specified string, array, or tuple.		
	op_index	Returns the element that corresponds to the index of a specified string, array, or tuple.		
	op_add	Calculates the sum of multiple values. The values can be strings or numbers.		
	op_max	Returns the largest value among the values of multiple fields or expressions.		
	op_min	Returns the smallest value among the values of multiple fields or expressions.		
	ct_int	Converts the value of a field or an expression to an integer.		
	ct_float	Converts the value of a field or an expression to a floating-point number.		
	ct_str	Converts the value of a field or an expression to a string.		
	ct_bool	Converts the value of a field or an expression to a Boolean value.		
	ct_chr	Converts the ANSI or Unicode value of a field or an expression to a character.		
Conversion functions	ct_ord	Converts the value of a field or an expression to an ANSI value or a Unicode value.		
	ct_hex	Converts the value of a field or an expression to a hexadecimal number.		
	ct_oct	Converts the value of a field or an expression to an octal number.		
	ct_bin	Converts the value of a field or an expression to a binary number.		
	bin2oct	Converts a binary byte string to an octal string.		
	bin2hex	Converts a binary byte string to a hexadecimal string.		
	op_abs	Returns the absolute value of an input value.		
	op_div_floor	Returns the integer part of the quotient of two input values.		
	op_div_true	Returns the quotient of two input values.		
	op_pow	Returns an input value raised to a specified power.		
	op_mul	Returns the product of two input values.		
	op_neg	Returns the opposite number of an input value.		
	op_mod	Returns the remainder of an input value divided by the other input value.		
	op_sub	Returns the difference between two input values.		
	op_round	Returns an input value rounded.		
	op_sum	Returns the sum of input values.		
	mat_ceil	Rounds an input value rounded up to the nearest integer.		
	mat_exp	Returns Euler's number raised to the power of an input value.		
	mat_fabs	Returns the absolute value of an input value.		
	mat_floor	Rounds an input value down to the nearest integer.		
Arithmetic functions	mat_log	Returns the logarithm of an input value with the base specified by the other input value.		
	mat_log10	Returns the base-10 logarithm of an input value.		

	mat_sqrt	Returns the square root of an input value.	
	mat_degrees	Converts radians to degrees.	
	mat_radians	Converts degrees to radians.	
	mat_sin	Returns the sine of an input value in radians.	
	mat_cos	Returns the cosine of an input value in radians.	
	mat_tan	Returns the tangent of an input value in radians.	
	mat_acos	Returns the arc cosine of an input value in radians.	
	mat_asin	Returns the arc sine of an input value in radians.	
	mat_atan	Returns the arc tangent of an input value in radians.	
	mat_atan2	Returns the arc tangent of X and Y coordinates.	
	mat_atanh	Returns the inverse hyperbolic tangent of an input value.	
	mat_hypot	Returns the Euclidean norm of two input values.	
	str_format	Formats strings.	
	str_join	Concatenates input strings to generate a new string by using a specified connector.	
	str_zip	Concurrently splits two values or strings that are returned by expressions and combines the results into one string.	
	str_encode	Encodes a string by using a specified encoding format.	
	str_decode	Decodes an input value by using a specified encoding format.	
	str_hex_escape_encode	Escapes special characters. The function can escape hexadecimal characters to Chinese characters.	
	str_sort	Sorts a specified object.	
	str_reverse	Reverses a string.	
	str_replace	Replaces an existing string with a specified string based on a specified rule.	
	str_logtash_config_normaliz e	Converts data in the Logstash configuration language to the JSON format.	
	str_translate	Replaces specified characters in a string with mapping characters.	
	str_strip	Deletes specified characters from a string.	
	str_lstrip	Deletes specified characters from the start of a string.	
	str_rstrip	Deletes specified characters from the end of a string.	
	str_lower	Converts all uppercase letters in a string to lowercase letters.	
	str_upper	Converts all lowercase letters in a string to uppercase letters.	
	str_title	Capitalizes the first letter of each word in a string and converts the other letters in the string to lowercase letters.	
	str_capitalize	Capitalizes the first letter of a string and converts the other letters in the string to lowercase letters.	
	str_swapcase	Converts the uppercase letters to lowercase letters and lowercase letters to uppercase letters in a string.	
	str_count	Counts the number of occurrences of a character in a string.	
	str_find	Checks whether a string contains a specified substring.	
	str_rfind	Returns the position of the last occurrence of a specified character in a string.	
String functions	str_endswith	Checks whether a string ends with a specified suffix.	
	str_startswith	Checks whether a string starts with a specified string.	
	str_split	Splits a string by using a specified delimiter.	
	str_splitlines	Splits a string by using a line feed.	
	str_partition	Splits a string into three parts from left to right by using a specified delimiter.	
	str_rpartition	Splits a string into three parts from right to left by using a specified delimiter.	
	str_center	Pads a string to a specified length by using a specified character.	
	str_ljust	Pads a string to a specified length by using a specified character from the end of the string.	
	str_rjust	Pads a string to a specified length by using a specified character from the start of the string.	
	str_zfill	Pads a string to a specified length by using 0 from the start of the string.	
	str_expandtabs	Converts \t in a string to spaces.	
	str_isalnum	Checks whether a string contains only letters and digits.	

User Guide-Log Service

	str_isalpha	Checks whether a string contains only letters.		
	str_isascii	Checks whether a string is in the ASCII table.		
	str_isdecimal	Checks whether a string contains only decimal characters.		
	str_isdigit	Checks whether a string contains only digits.		
	str_isidentifier	Checks whether a string is a valid Python identifier or checks whether a variable name is valid.		
	str_islower	Checks whether a string contains lowercase letters.		
	str_isnumeric	Checks whether a string contains digits.		
	str_isprintable	Checks whether all characters in a string are printable characters.		
	str_isspace	Checks whether a string contains only spaces.		
	str_istitle	Checks whether the first letter of each word in a string is in uppercase and the other letters in the string are in lowercase.		
	str_isupper	Checks whether all letters in a string are in uppercase.		
	str_uuid	Generates a random universally unique identifier (UUID).		
	dt_parse	Converts a value or the value of a time expression to a datetime object.		
	dt_str	Converts a value or the value of a time expression to a string.		
	dt_parsetimestamp	Converts a value or the value of a time expression to a UNIX timestamp.		
	dt_prop	Returns a specific attribute of a value, or returns a specific attribute of the value of a time expression. The attribute can be day or year.		
	dt_now	Returns the current date and time.		
	dt_today	Return only the current date.		
	dt_utcnow	Returns the current datetime object in the current time zone.		
	dt_fromtimestamp	Converts a UNIX timestamp to a datetime object.		
	dt_utcfromtimestamp	Converts a UNIX timestamp to a datetime object in the current time zone.		
	dt_strptime	Parses a time string into a datetime object.		
	dt_currentstamp	Returns the current UNIX timestamp.		
	dt_totimestamp	Converts a datetime object to a UNIX timestamp.		
	dt_strftime	Converts a datetime object to a string in a specified format.		
	dt_strftimestamp	Converts a UNIX timestamp to a string in a specified format.		
Date and time functions	dt_truncate	Extracts a time value from a value or the value of a time expression based on a specified time granularity.		
	dt_add	Changes a value or the value of a time expression based on a specified time granularity.		
	dt_MO	Offsets a specified time to the same date of the previous or next Nth Monday. The offset value N is passed to the weekday parameter of the dt_add function.		
	dt_TU	Offsets a specified time to the date of the previous or following Nth Tuesday. The offset value N is passed to the weekday parameter of the dt_add function.		
	dt_WE	Offsets a specified time to the date of the previous or following Nth Wednesday. The offset value N is passed to the weekday parameter of the dt_add function.		
	dt_TH	Offsets a specified time to the date of the previous or following Nth Thursday. The offset value N is passed to the weekday parameter of the dt_add function.		
	dt_FR	Offsets a specified time to the date of the previous or following Nth Friday. The offset value N is passed to the weekday parameter of the dt_add function.		
	dt_SA	Offsets a specified time to the date of the previous or following Nth Saturday. The offset value N is passed to the weekday parameter of the dt_add function.		
	dt_SU	Offsets a specified time to the date of the previous or following Nth Sunday. The offset value N is passed to the weekday parameter of the dt_add function.		
	dt_astimezone	Converts a value or the value of a time expression to a datetime object in a specified time zone.		
	dt_diff	Returns the difference between two values or between the values of two time expressions based on a specified time granularity.		
	regex_select	Extracts a value that matches a regular expression.		
	regex_findall	Extracts all values that match a regular expression.		
Regular expression functions	regex_match	Checks whether a value matches a regular expression.		
	regex_replace	Replaces the characters that match a regular expression in a string.		
	regex_split	Splits a string into an array of strings.		

Cloud Defined Storage

Grok function	grok	Extracts a value that matches a regular expression.		
Structured data functions	json_select	Extracts or calculates specific values from a JSON expression by using JMESPath.		
	json_parse	Parses a value into a JSON object.		
	xml_to_json	Converts XML data to JSON data, and then expands the converted data.		
	geo_parse	Identifies the city, province, and country based on an IP address.		
	ip_cidrmatch	Checks whether an IP address belongs to a Classless Inter-Domain Routing (CIDR) block.		
	ip_version	Checks whether the version of an IP address is IPv4 or IPv6.		
	ip_type	Identifies the type of an IP address and checks whether the type of the IP address is private or public.		
IP address parsing functions	ip_makenet	Converts an IP address to a CIDR block.		
	ip_to_format	Converts the format of a CIDR block to a format that specifies the netmask or prefix length of the CIDR block.		
	ip_overlaps	Checks whether two CIDR blocks overlap.		
	ip2long	Converts an IP address to a value of the long type.		
	long2ip	Converts a value of the long type to an IP address.		
	url_encoding	Encodes URL data.		
	url_decoding	Decodes URL data.		
	base64_encoding	Encodes data by using the Base64 algorithm.		
	base64_decoding	Decodes data by using the Base64 algorithm.		
	html_encoding	Encodes data in the HTML format.		
	html_decoding	Decodes HTML-encoded data.		
	md5_encoding	Encodes data by using the MD5 algorithm.		
Encoding and decoding	sha1_encoding	Encodes data by using the SHA1 algorithm.		
functions	gzip_compress	Compresses and encodes data.		
	gzip_decompress	Decompresses compressed data.		
	zlib_compress	Compresses and encodes data.		
	zlib_decompress	Decompresses compressed data.		
	aes_encrypt	Encrypts data by using the AES algorithm.		
	aes_decrypt	Decrypts data by using the AES algorithm.		
	jwt_encoding	Encodes JSON data based on the JWT standard.		
	jwt_decoding	Decodes data to raw JSON data based on the JWT standard.		
	ua_parse_device	Parses User-Agent and returns the device information.		
Parsing functions	ua_parse_os	Parses User-Agent and returns the operating system information.		
	ua_parse_agent	Parses User-Agent and returns the browser information.		
	ua_parse_all	Parses User-Agent and returns all information.		
	lst_make	Constructs a list.		
	lst_insert	Inserts elements to a specified position in a list.		
List functions	lst_append	Appends elements to a list.		
	lst_delete_at	Deletes the element at a specified position from a list.		
	lst_reverse	Reverses the order of elements in a list.		
	lst_get	Returns the element at a specified position in a list or a tuple.		
	dct_make	Constructs a dictionary.		
	dct_update	Updates a dictionary.		
Dictionary functions	dct_delete	Deletes key-value pairs from a dictionary.		
	dct_keys	Returns the keys of a dictionary.		
	dct_values	Returns the values of a dictionary.		
	dct_get	Returns the value that corresponds to a specified key in a dictionary.		
- 11 A	tab_parse_csv	Constructs a table from CSV-formatted text.		
Table functions	tab_to_dict	Constructs a dictionary from a table.		

Parsing functions	res_local	Pulls the values of advanced parameters from the current data transformation job.
	res_rds_mysql	Pulls data from a specified table in a database that is created on an ApsaraDB RDS for MySQL instance or obtains the execution result of an SQL statement. The data and result can be updated at regular intervals.
	res_log_logstore_pull	Pulls data from another Logstore when you transform data in a Logstore. You can pull data in a continuous manner.
	res_oss_file	Pulls data from an object in a specified Object Storage Service (OSS) bucket. The data can be updated at regular intervals.

4.5.8.6. Global processing functions

4.5.8.6.1. Overview of global processing functions

This topic describes all global operation functions that are provided by Log Service. The domain-specific language (DSL) for Log Service provides approximately 30 global processing functions. You can use the functions to control the logic of data transformation. The following table describes the global processing functions.

Category	Function	Description
	e_if	Performs an operation if a specified condition is met. You can specify multiple condition-operation pairs.
	e_if_else	Performs an operation based on the evaluation result of a condition.
Flow control functions	e_switch	Performs an operation if a specified condition is met. You can specify multiple condition-operation pairs. Performs an operation if a specified condition is met and returns the result.
	e_compose	Combines multiple operations and performs the operations in sequence.
	e_drop	Discards a log if a specified condition is met.
	e_keep	Retains a log if a specified condition is met.
Event processing functions	e_split	Splits a log into multiple logs based on the value of a specified field. You can also use the JMESPath expression to extract the value of the field, and then split the log.
	e_output	Writes a log to a specified Logstore. The subsequent transformation rules are not executed for the log.
	e_coutput	Writes a log to a specified Logstore. The subsequent transformation rules are still executed for the log.
	e_to_metric	Converts logs to metrics that can be stored in a Metricstore.
	v	Extracts the value of a field from a log.
	e_set	Adds a field or specifies a new value for an existing field.
	e_drop_fields	Deletes the log fields that meet a specified condition.
Field processing functions	e_keep_fields	Retains the log fields that meet a specified condition.
	e_pack_fields	Encapsulates log fields and assigns the log fields as a value to a new field.
	e_rename	Renames the log fields that meet a specified condition.
	e_regex	Extracts the value of a field based on a regular expression and assigns the value to other fields.
	e_json	Performs operations on JSON objects in a specified field. You can configure the parameters to expand JSON data, extract JSON data by using the JMESPath expression, or expand the extracted JSON data.
	e_kv	Extracts key-value pairs from multiple input fields by using a specified quote.
	e_kv_delimit	Extracts key-value pairs from input fields by using a specified delimiter.
	e_csv	Extracts multiple fields by using a default delimiter. The default delimiter is a comma (,).
Value extraction functions	e_tsv	Extracts multiple fields by using a default delimiter. The default delimiter is a tab (\t).
	e_psv	Extracts multiple fields by using a default delimiter. The default delimiter is a vertical bar ().

	e_syslogrfc	Calculates the values of the facility and severity fields and returns the value of the facilitylabel field that indicates level information. The function calculates the values based on the value of the priority field and the specified syslog protocol.
	e_anchor	Extracts strings by using the rules specified by anchor_rules.
Mapping and enrichment functions	e_dict_map	Maps the value of an input field to a value in a specified dictionary and returns a new field.
	e_table_map	Maps the value of an input field to a row in a specified table and returns a new field.
	e_search_dic_map	Searches the keywords in a specified dictionary for a raw log field, maps the field to a value in the dictionary, and returns a new field. The keywords must be query strings.
	e_search_table_map	Searches a specified column in a specified table for a raw log field, maps the field to a row in the table, and returns a new field. The values of the column must be query strings.
Value-added content function	e_threat_intelligence	Obtains the threat intelligence for an IP address or a domain name that is specified by a log field and assigns the threat intelligence as a value to a specified field.

4.5.8.6.2. Flow control functions

This topic describes the syntax and parameters of flow control functions. This topic also provides examples on how to use the functions.

Functions

Function	Description
e_compose	 Combines multiple operations. The function is commonly used in the e_if, e_switch, or e_if_else function. The function performs specified operations on a log in sequence and returns the result. If the function performs an operation that deletes a log, the function no longer performs other operations on the log.
e_if	Performs an operation if a specified condition is met. You can specify multiple condition-operation pairs. If a condition is met, the function performs the operation that corresponds to the condition. If the condition is not met, the function does not perform the operation, but evaluates the next condition. If the function performs an operation that deletes a log, the function no longer performs other operations on the log. e_if(e_has("a"), e_output("target-a"), e_has("b"), e_output("target-b"), The preceding example of the function is equivalent to the following Python code: if e_has("a"): e_output("target-a") if e_has("b"): e_output("target-b")
e_if_else	<pre>Performs an operation based on the evaluation result of a condition. e_if_else(e_has("a"), e_output("target-a"), e_output("target-b")) The preceding example of the function is equivalent to the following Python code: if e_has("a"): e_output("target-a") else: e_output("target-b")</pre>

	 Performs an operation if a specified condition is met. You can specify multiple condition-operation pairs. If a condition is met, the function performs the operation that corresponds to the condition and returns the result. If the condition is not met, the function does not perform the operation, but evaluates the next condition. If no specified conditions are met and the default parameter is configured, the function performs the operation that is specified by the default parameter and returns the result. If the function performs an operation that deletes a log, the function no longer performs other operations on the log.
e_switch	<pre>e_switch(e_has("a"), e_output("target-a"), e_has("b"), e_output("target-b"), default=e_output("target-default"),)</pre>
	<pre>The preceding example of the function is equivalent to the following Python code: if e_has("a"): e_output("target-a") elif e_has("b"): e_output("target-b") else: e_output("target-default")</pre>

e_compose

The e_compose function combines multiple operations.

Syntax

e_compose(operation_1, operation_2, ...)

• Parameters

Parameter	Туре	Required	Description
operation_1	Global processing function	Yes	A global processing function or a combination of global processing functions.
operation_2	Global processing function	No	A global processing function or a combination of global processing functions.

Response

A log on which the specified operations are performed is returned.

Example

If the value of the content field is 123, delete the age and name fields and then change the value of the content field to ctx.

```
• Raw log
content: 123
age: 23
name: twiss
```

• Transformation rule

```
e_if(
    e_search("content==123"),
    e_compose(e_drop_fields("age|name"), e_rename("content", "ctx")),
)
o Result
```

ctx: 123

e_if

The e_if function performs an operation if a specified condition is met.

```
• Syntax
```

```
e_if(condition, operation)
e_if(condition_1, operation_1, condition_2, operation_2, ...)
```

O Note You must specify the condition and operation parameters in pairs.

• Parameters

Parameter	Туре	Required	Description
condition	Arbitrary	Yes	An expression or a combination of expressions. If the result is not a Boolean value, the system evaluates whether the condition is true or false.
operation	Global processing function	No	A global processing function or a combination of global processing functions.

Response

A log on which the specified operations are performed is returned.

Examples

• Example 1: Match a field value against specified values and perform an operation. If the value of the **result** field is failed or **failure**, set the **__topic__** field to <code>__login_failed_event</code> .

e_if(e_match("result", r"failed|failure"), e_set("__topic__", "login_failed_event"))

• Example 2: Perform evaluation based on a field value and perform an operation. If the request_body field exists and the field is not empty, call the field processing function e_json to expand the value of the request_body field into multiple values.

e_if(v("request_body"), e_json("request_body"))

• Example 3: Perform advanced evaluation and perform an operation. If the value of the **valid** field is failed in lowercase, discard the log.

e_if(op_eq(str_lower(v("valid")), "failed"), DROP)

- Example 4: Perform multiple operations in sequence based on specified conditions.

 - e_if(True, e_set("__topic__", "default_login"),
 e_match("valid", "failed"), e_set("__topic__", "login_failed_event")

e_if_else

The e_if_else function performs an operation based on the evaluation result of a specified condition.

Syntax

e_if_else(condition, operation_1 if the condition evaluates to true, operation_2 if the condition evaluates to false)

• Parameters

Parameter	Туре	Required	Description
condition	Arbitrary	Yes	An expression or a combination of expressions. If the result is not a Boolean value, the system evaluates whether the condition is true or false.
operation_1 if the condition evaluates to true	Global processing function	Yes	A global processing function or a combination of global processing functions.
operation_2 if the condition evaluates to false	Global processing function	Yes	A global processing function or a combination of global processing functions.

```
    Response
```

A log on which an operation is performed based on the evaluation result of the specified condition is returned.

Example

If the value of the **result** field is ok or pass or if the value of the **status** field is 200, retain the log.

```
    Raw logs
```

```
result: ok
     status: 400
     result: Pass
     status: 200
     result: failure
     status: 500
 • Transformation rule
     e_if_else(
         op_or(e_match("result", r"(?i)ok|pass"), e_search("status== 200")), KEEP, DROP
     )
 • Result: The first two logs are retained. The third log is discarded.
     result: ok
     status: 400
     result: Pass
     status: 200
e switch
The e_switch function performs an operation if a specified condition is met.

    Syntax

   e_switch(condition_1, operation_1, ..., default=None)
    ? Note You must specify the condition and operation parameters in pairs.

    Parameters

                                                               Required
   Parameter
                                 Type
                                                                                             Description
                                                                                             An expression or a combination of expressions. If the result is
```

Yes

Arbitrary

condition

not a Boolean value, the system evaluates whether the condition is true or false.

operation	Global processing function	Yes	A global processing function or a combination of global processing functions.
default	Global processing function	No	A global processing function or a combination of global processing functions. If no specified conditions are met, the operation specified by the default parameter is performed.

Response

A log on which the specified operations are performed is returned.

- Example
- If the value of the content field is 123, set the _topic_ field to Number. If the value of the data field is 123, set the _topic_ field to PRO.
- Raw logs

```
__topic__:
age: 18
content: 123
name: maki
data: 342
__topic__:
```

age: 18 content: 23 name: maki data: 123

Transformation rule

e_switch(

```
e_search("content==123"),
e_set("_topic_", "Number", mode="overwrite"),
e_search("data==123"),
e_set("_topic_", "PRO", mode="overwrite"),
)
```

Result

```
__topic__: Number
age: 18
content: 123
name: maki
data: 342
__topic__: PRO
```

age: 18 content: 23 name: maki data: 123

- You can combine the e_switch and e_output functions to ship the logs that meet specified conditions to different Logstores. If you specify default=e_drop(), the logs that do not meet specified conditions are discarded and not shipped. If you do not configure the default parameter, the logs that do not meet specified conditions are shipped to the first Logstore that you specify.
 - Raw logs

```
_topic_: sas-log-dns
test: aliyun
__topic_: aegis-log-network
test:ecs
__topic_: local-dns
test:sls
__topic_: aegis-log-login
test: sls
```

```
e_match("_topic_","sas-log-process"),e_output(name="target2"),
e_match("_topic_","sas-log-process"),e_output(name="target2"),
e_match("_topic_","local-dns"),e_output(name="target3"),
e_match("_topic_","aegis-log-network"),e_output(name="target4"),
e_match("_topic_","aegis-log-login"),e_output(name="target5"),
defaulte= drop())
```

4.5.8.6.3. Event processing functions

This topic describes the syntax and parameters of event processing functions. This topic also provides examples on how to use the functions.

Functions

Category	Function	Description
	e_drop	Discards a log if a specified condition is met.

Event processing	e_keep	<pre>Retains a log if a specified condition is met. Both the e_keep and e_drop functions can be used to discard logs. The difference is that the e_keep function discards logs if a specified condition is not met, whereas the e_drop function discards logs if a specified condition is met. # The following transformation rules are equivalent: e_if_else(e_search("fl==v1"), KEEP, DROP) e_if_else(e_search("not fl==v1"), DROP) e_keep(e_search("fl==v1")) e_drop(e_search("not fl==v1")) # The following transformation rules are invalid: e_if(e_search(""), KEEP) e_keep()</pre>
Event splitting	e_split	Splits a log into multiple logs based on the value of a specified field. You can also use the JMESPath expression to extract the value of the field, and then split the log.
Event generation	e_output and e_coutput	 Writes a log to a specified Logstore. You can specify the topic, source, tags and MD5 hash keys of a shard for the log. e_output: writes a log to a specified Logstore. The subsequent transformation rules are not executed for the log. e_coutput: writes a log to a specified Logstore. The subsequent transformation rules are executed for the log.

e_drop

The e_drop function discards a log if a specified condition is met.

Syntax

e_drop(condition=True)

The identifier **DROP** is supported. The identifier DROP is equivalent to the **e_drop()** function.

• Parameters

Parameter	Туре	Required	Description
condition	Bool	No	Default value: True. In most cases, one condition is passed to a function.

Response

If the specified condition is met, the log is discarded and None is returned. If the specified condition is not met, the log is returned.

• Examples

- Example 1: If the value of the _programe_ field in a log is access, discard the log. Otherwise, retain the log.
 - Raw log
 __programe__: access
 age: 18
 content: 123
 name: maki
 __programe__: error
 age: 18
 content: 123
 name: maki

Transformation rule

e_if(e_search("__programe__=access"), DROP)

```
    Result
```

The log in which the value of the **_programe_** field is access is discarded. The log in which the value of the **_programe_** field is error is retained.

__programe__: error age: 18 content: 123 name: maki

• Example 2: If the specified condition evaluates to True, discard the log.

Raw log

k1: v1 k2: v2

k3: k1

Transformation rule

e_drop(e_search("k1==v1"))

Result

The log is discarded because the k1 = v1 condition evaluates to True.

- Example 3: If the specified condition evaluates to False, retain the log.
 - Raw log
 - k1: v1 k2: v2 k3: k1
 - Transformation rule

e_drop(e_search("not k1==v1"))

- Result
 - k1: v1 k2: v2 k3: k1

• Example 4: If no conditions are specified, use True, which indicates that the log is discarded.

- Raw log
 - k1: v1 k2: v2 k3: k1
- Transformation rule
- e_drop()
- Result
 - The log is discarded.

e_keep

The e_keep function retains a log if a specified condition is met.

Syntax

e_keep(condition=True)

The identifier **KEEP** is supported. The identifier KEEP is equivalent to the **e_keep()** function.

• Parameters

Parameter	Туре	Required	Description
condition	Bool	No	Default value: True. In most cases, one condition is passed to a function.

- Response
- If the specified condition is met, the log is returned. If the specified condition is not met, the log is discarded.
- Examples
 - Example 1: If the value of the __programe__ field in a log is access, retain the log. Otherwise, discard the log.
 - Raw log

```
__programe__: access
age: 18
content: 123
name: maki
__programe__: error
age: 18
content: 123
name: maki
```

Transformation rule

```
e_keep(e_search("__programe__==access"))
# Equivalent to:
e_if(e_search("not __programe__==access"), DROP)
# Equivalent to:
e_if_else(e_search("__programe__==access"), KEEP, DROP)
```

Result

The log in which the value of the __programe__ field is access is retained.

```
__programe__: access
age: 18
content: 123
name: maki
```

• Example 2: If the specified condition evaluates to True, retain the log.

Raw log

- k1: v1 k2: v2 k3: k1
- Transformation rule

e_keep(e_search("k1==v1"))

- Result
 - k1: v1 k2: v2 k3: k1
- Example 3: If the specified condition evaluates to False, discard the log.
 - Raw log
 - k1: v1 k2: v2
 - k3: k1
 - Transformation rule

e_keep(e_search("not k1==v1"))

Result

The log is discarded.

- Example 4: Pass the value False to the e_keep function.
 - Raw log
 - k1: v1 k2: v2 k3: k1
 - Transformation rule

e_keep(False)

- Result
- The log is discarded.

e_split

The e_split function splits a log into multiple logs based on the value of a specified field. You can also use the JMESPath expression to extract the value of the field, and then split the log.

Syntax

e_split(Field name, sep=',', quote='"', lstrip=True, jmes=None, output=None)

Splitting rules:

- i. If you configure the **jmes** parameter, Log Service converts the values of the log field to a JSON list, uses the JMESPath expression to extract the values from the JSON list, and then uses these values in the next operation. If you do not configure the **jmes** parameter, Log Service uses the values of the log field in the next operation.
- ii. If the values obtained from the previous operation is a list or a string that represents a JSON list, Log Service splits the log based on the list. Otherwise, Log Service parses the values into CSV values based on the sep, quote, or Istrip parameter. Then, Log Service splits the log based on the parsed values.

[•] Parameters

Parameter	Туре	Required	Description
Field name	String	Yes	The name of the field that is used to split a log.
sep	String	No	The delimiter that is used to separate values.
quote	String	No	The quote that is used to enclose a value.
lstrip	String	No	Specifies whether to remove the spaces to the left of a value. Default value: True.
jmes	String	No	The JMESPath string that is used to convert the values of the field to a JSON object and extract the values from the JSON object. Log Service splits the log based on the JSON object.
output	String	No	The new name of the field, which overwrites the existing name by default.

Response

A list of logs is returned. The values of fields in the returned logs are the same as the values of the fields in the raw log.

Example

Raw log

__topic__: age: 18 content: 123 name: maki

__topic__: age: 18 content: 123 name: maki

• Transformation rule

e_set("__topic__", "V_SENT,V_RECV,A_SENT,A_RECV")
e_split("__topic__")

Result

```
__topic__: A_SENT
age: 18
content: 123
name: maki
__topic__: V_RECV
age: 18
content: 123
name: maki
```

e_output and e_coutput

The e_output and e_coutput functions write a log to a specified Logstore. You can specify the topic, source, and tags for the log.

Syntax

e_output(name=None, project=None, logstore=None, topic=None, source=None, tags=None, hash_key_field=None, hash_key=None)
e_coutput(name=None, project=None, logstore=None, topic=None, source=None, tags=None, hash_key_field=None, hash_key=None)

During preview, the log is written to a Logstore named internal-etl-log instead of the specified Logstore. The first time that you preview data transformation results, Log Service automatically creates a dedicated Logstore named internal-etl-log in the current project. You cannot modify the configurations of this Logstore or write other data to the Logstore. You are not charged for this Logstore.

Parameters

③ Note If you configure the name, project, and logstore parameters in the e_output or e_coutput function and specify the project and Logstore in the **Create Data Transformation Rule** panel, the configurations in the e_output or e_coutput function take precedence. The following list describes the configurations:

 If you configure only the name parameter in the e_output or e_coutput function, the transformation result is sent and stored in the Logstore that corresponds to the name parameter.

 If you configure only the project and logstore parameters in the e_output function, the transformation result is sent and stored in the Logstore specified in the e_output function.

If you use an AccessKey pair to authorize data transformation, the AccessKey pair of the current logon account is used to transform data.

 If you configure the name, project, and logstore parameters in the e_output function, the transformation result is sent and stored in the Logstore specified in the e_output function.

If you use an AccessKey pair to authorize data transformation, the AccessKey pair specified in the storage destination is used to transform data.

Parameter	Туре	Required	Description
name	String	No	The name of the storage destination. Default value: None.
project	String	No	The existing project to which the log is written.
logstore	String	No	The existing Logstore to which the log is written.
topic	String	No	The new topic of the log.
source	String	No	The new source of the log.
tags	Dict	No	The new tags of the log. The tags are in the dictionary format. ⑦ Note You do not need to prefix keywords with tag: .
hash_key_field	String	No	The name of the field that is used for hashing. The log is written to a shard of the storage destination that you specify based on the hash value of the field. Note If the log does not contain the field that you specify, the log is randomly written to a shard of the storage destination that you specify in load balancing mode.

			The hash value. The log is written to a shard of the storage destination that you specify based on the hash value. (2) Note The hash_key_field parameter has a higher
hash_key	String	No	priority than the hash_key parameter. If the hash_key_field parameter is configured in a transformation rule, the hash_key parameter does not take effect.

· Default storage destination

To use the e_output or e_coutput function, you must configure a default storage destination in the **Create Data Transformation Rule** panel. By default, Log Service uses the storage destination labelled 1 as the default storage destination. In the following figure, the transformation result is shipped to the Logstores that corresponds to target_01, target_02, and target_03. Data that is not discarded during transformation is stored in the Logstore that corresponds to the default storage destination named target_00.

Advanced Parameter Settings

If the project or Logistore that you specify in the e_output or e_coutput function does not exist, you can specify key-value pairs in the Advanced Parameter Settings section of the Create Data Transformation Rule panel. You can set a key to config.sls_output.failure_strategy and the value of the key to {"drop_when_not_exists":"true"} to skip logs. The skipped logs are discarded and reported as warning logs. If you do not specify key-value pairs in the Advanced Parameter Settings section, your data transformation job is suspended until the project or Logstore that you specify is created.

Warning If the specified project or Logstore does not exist and you specify key-value pairs in the **Advanced Parameter Settings** section to skip logs, the skipped logs are discarded. Proceed with caution.

• Result

- e_output: writes a log to a specified Logstore. The subsequent transformation rules are not executed for the log.
- e_coutput: sends log entries to the specified Logstore. After a log entry is sent, the remaining transformation rules are still executed on the log entry.
- Examples

• Example 1: Evaluate the value of the k2 field in a log against the regular expression. If the value meets the regular expression, write the log to the Logstore specified in target2 and set topic to topic1.

Raw log

__topic__: k1: v1 k2: v2 x1: v3 x5: v4

Transformation rule

The $e_drop()$ function deletes the data that does not meet the condition of the $e_dif()$ function. If you do not add the $e_drop()$ function to the transformation rule, the data that does not meet the condition of the $e_df()$ function is shipped to the default storage destination.

e_if(e_match("k2", r"\w+"), e_output(name="target2", source="source1", topic="topic1"))
e_drop()

Result

__topic__: topic1 k1: v1 k2: v2 x1: v3 x5: v4

• Example 2: Calculate the hash value of a log based on the value of the **db_version** field and write the log to a shard of the storage destination that you specify based on the hash value.

Raw log

__topic__: db_name: db-01 db_version:5.6 __topic__: db_name: db-02

db_version:5.7

Transformation rule

e_output(name="target1", hash_key_field="db_version")

Result

- # For example, the storage destination named target1 has two shards.

The hash values for logs whose values of the db_version field are 5.6 and 5.7 are 0ebela34e990772a2bad83ce076e0766 and f1867131d82f2256b4521fe34aec2405.

Shard 0: __topic__: db_name: db-01 db_version:5.6

Shard 1: __topic__: db_name: db-02 db_version:5.7 • Example 3: Specify the hash value for a log and write the log to a shard of the storage destination that you specify based on the hash value.

Raw log

__topic__: db_name: db-01 db_version:5.6

__topic__: db_name: db-02 db version:5.7

Transformation rule

Result

```
# For example, the storage destination has two shards.
```

```
# Shard 0:
__topic_:
db_name: db-01
db version:5.6
_topic_:
```

db_name: db-02 db_version:5.7 # Shard 1:

None

4.5.8.6.4. Field processing functions

This topic describes the syntax and parameters of field processing functions. This topic also provides examples on how to use the functions.

Functions

Function	Description
v	Extracts the value of a field from a log. If you specify the names of multiple fields for the function, the function returns the value of the first field that exists in the log.
e_set	Adds a field or specifies a new value for an existing field.
e_drop_fields	Deletes the log fields that meet a specified condition.
e_keep_fields	Retains the log fields that meet a specified condition.
e_pack_fields	Encapsulates log fields and assigns the log fields as a value to a new field.
e_rename	Renames the log fields that meet a specified condition.

v

The v function extracts the value of a field from a log. If you specify the names of multiple fields for the function, the function returns the value of the first field that exists in the log.

Syntax

v(key, ..., default=None)

Parameters

Parameter	Туре	Required	Description
key	String	Yes	The name of the field.
default	Arbitrary	No	If the field does not exist, the function returns the value of this parameter. Default value: None.

Response

The value of the first field that exists in the log is returned. If the field does not exist, the value of the default parameter is returned.

• Example

- Assign the value of the content field to the test_content field.
- Raw log

content: hello

Transformation rule

e_set("test_content", v("content"))

• Result

content: hello

test_content: hello

e_set

The e_set function adds a field or specifies a new value for an existing field.

Svntax

e_set(key1, value1, key2, value2, mode="overwrite")

() Important

- You must specify the key1 and value1 parameters in pairs.
- If you use the e_set function to specify a value for a time field, such as F_TIME or __time__, the value must be a numeric string.

• Parameters

Parameter	Туре	Required	Description
key	String	Yes	The name of the new field that you want to add or the name of the existing field for which you want to specify a new value. You can obtain a name based on a string expression.
value	Arbitrary	Yes	The value of the new field or the new value of the existing field. If the value of this parameter is not a string, the function automatically converts the value to a string. For example, if you set this parameter to a value of the tuple, list, or dictionary type, the function automatically converts the value to a JSON string. Note If you set this parameter to None, the function returns the raw log.
mode	String	No	The overwrite mode of fields. Default value:overwrite. For more information, see Field extraction check and overwrite modes

- Response
- The updated log is returned.
- Examples
- Example 1: Specify a fixed value for a field.
 Add a new field named city and set the value to Shanghai.

e_set("city", "Shanghai")

Example 2: Extract the value of an existing field and assign the value to a new field.
 Call an expression function to extract the value of an existing field named ret and assign the value to a new field named result.

e_set("result", v("ret"))

- Example 3: Specify a dynamic value for a field.
- Call multiple expression functions in sequence to obtain the lowercase value of the first field that exists and assign the value to the result field. e_set("result", str_lower(v("ret", "return")))
- Example 4: Specify different values for a field.
- a. Specify a value for the **event_type** field.

e_set("event_type", "login event", "event_info", "login host")

b. If the value of the ret field is fail, set the event type field to login failed event.

e_if(e_search('ret==fail'), e_set("event_type", "login failed event"))

e_drop_fields

The e_drop_fields function deletes the log fields that meet a specified condition.

Syntax

e_drop_fields(key1, key2, ...,regex=False)

Parameters

Parameter	Туре	Required	Description
key	String	Yes	The name of the log field. The value of this parameter can be a regular expression. If the field name meets the specified condition, the field is deleted. Otherwise, the field is retained. For more information about regular expressions, see Regular expressions. You must specify at least one log field.
regex	Boolean	No	If you set this parameter to False, regular expressions are not used for matching. Default value: True.

Response

The log from which the field is deleted is returned.

Example

If the value of the **content** field is 123, delete the **content** and **age** fields.

Raw log

age: 18 content: 123 name: twiss

• Transformation rule

e_if(e_search("content==123"), e_drop_fields("content", "age",regex=True))

• Result

name: twiss

e_keep_fields

The e_keep_fields function retains the log fields that meet a specified condition.

⑦ Note Log Service provides built-in meta fields, such as __time__ and __topic__. If you do not retain the __time__ field when you call the e_keep_fields function, the log time is reset to the current system time. If you do not want to reset the value of a meta field, you must add the meta field to a list in the F_TIME, F_META, F_TAGS, "f1", "f2" format. For more information, see Fixed identifiers.

Syntax

e_keep_fields(key1, key2,,regex=False)

• Parameters

Parameter	Туре	Required	Description
key	String	Yes	The name of the log field. The value of this parameter can be a regular expression. If the field name meets the specified condition, the field is retained. Otherwise, the field is deleted. You must specify at least one log field.
regex	Boolean	No	If you set this parameter to False, regular expressions are not used for matching. Default value: True.

Response

The log in which the field is retained is returned.

Example

If the value of the **content** field is 123, retain the **content** and **age** fields.

Raw log

age: 18 content: 123 name: twiss

• Transformation rule

e_if(e_search("content==123"), e_keep_fields("content", "age"))

• Result

```
age: 18
content: 123
```

e_pack_fields

The e_pack_fields function encapsulates log fields and assigns the log fields as a value to a new field.

Syntax

e_pack_fields(output_fields,include=".*",exclude=None,drop_packed=True)

Parameters

Parameter	Туре	Required	Description
output_field	String	Yes	The name of the output field. The value of the output field is in the JSON format.
include	String	No	The whitelist. Fields that match the regular expression specified in the whitelist are encapsulated. Default value: ".*". This value indicates that all fields in a log are matched and encapsulated. For more information, see Regular expressions .
exclude	String	No	The blacklist. Fields that match the regular expression specified in the blacklist are not encapsulated. Default value: None. This value indicates that no fields in a log are evaluated. For more information, see Regular expressions.
drop_packed	Boolean	No	 Specifies whether to delete raw fields after the fields are encapsulated. Default value: True. True (default): The raw fields that are encapsulated are deleted in the result. False: The raw fields that are encapsulated are not deleted in the result.

Response

The log in which the fields are encapsulated is returned.

Cloud Defined Storage

• Examples

• Example 1: Encapsulate all log fields into the test field. By default, the raw fields that are encapsulated are deleted in the result.

```
    Raw log
```

- test1:123 test2:456 test3:789
- Transformation rule

e_pack_fields("test")

Result

test:{"test1": "123", "test2": "456", "test3": "789"}

- Example 2: Encapsulate all log fields into the test field. The raw fields that are encapsulated are not deleted in the result.
 - Raw log

test1:123 test2:456 test3:789

Transformation rule

e_pack_fields("test",drop_packed=False)

Result

```
test:{"test1": "123", "test2": "456", "test3": "789"}
test1:123
test2:456
test3:789
```

• Example 3: Encapsulate the test and abcd fields into the content field. The raw fields that are encapsulated are not deleted in the result.

```
    Raw log
```

```
abcd@#%:123
test:456
abcd:789
```

```
    Transformation rule
```

e_pack_fields("content", include="\w+", drop_packed=False)

Result

```
abcd:789
abcd@f%:123
content:{"test": "456", "abcd": "789"}
test:456
```

- Example 4: Encapsulate raw log fields that exclude the **test** and **abcd** fields into the **content** field. The raw fields that are encapsulated are deleted in the result.
 - Raw log

```
abcd@#%:123
test:456
abcd:789
```

Transformation rule

e_pack_fields("content", exclude="\w+", drop_packed=True)

Result

abcd:789 content:{"abcd@#%": "123"} test:456

e_rename

The e_rename function renames the log fields that meet a specified condition.

Syntax

e_rename("key1", "new key1", "key2", "new key2", ..., regex=False)

O Note You must specify the key and new key parameters in pairs.

Parameters

Parameter	Туре	Required	Description
key	String	Yes	The name of the log field. The value of this parameter can be a regular expression. If the field name meets the specified condition, the field is renamed. For more information about regular expressions, seeRegular expressions. You must specify at least one log field.
new key	String	Yes	The new name of the field.

User Guide-Log Service

•

.

regex	Boolean	No	If you set this parameter to False, regular expressions are not used for matching. Default value: True.
Response			
The renamed field is returned			
Examples			
 Example 1: Rename the fiel 	d host client_hos.		
 Raw log 			
host: 1006			
 Transformation rule 			
e_rename("host","client	_host")		
 Result 			
client_host: 1006			
Example 2: Do not renameRaw log	a log field if no fields meet the	specified condition.	
host: 1006			
 Transformation rule 			
e_rename("url","rename_	url")		
 Result 			
host: 1006			

4.5.8.6.5. Value extraction functions

This topic describes the syntax and parameters of value extraction functions. This topic also provides examples on how to use the functions.

Functions

Category	Function	Description
Extraction based on regular expressions	e_regex	Extracts the value of a field based on a regular expression and assigns the value to other fields.
Extraction based on JSON objects	e_json	Performs operations on JSON objects in a specified field. You can configure the parameters to expand JSON data, extract JSON data by using the JMESPath expression, or expand the extracted JSON data.
Extraction by using delimiters	e_csv, e_psv, and e_tsv	 Extracts multiple fields from a specified field by using a specified delimiter and predefined field names. e_csv: uses a comma (,) as the default delimiter. e_psv: uses a vertical bar () as the default delimiter. e_tsv: uses a tab (\t) as the default delimiter.
Extraction of key value pairs	e_kv	Extracts key-value pairs from multiple input fields by using a specified quote.
Extraction of key-value pairs	e_kv_delimit	Extracts key-value pairs from input fields by using a specified delimiter.
Extraction based on the syslog protocol	e_syslogrfc	Calculates the values of the facility and severity fields and returns the value of the facilitylabel field that indicates level information. The function calculates the values based on the value of the priority field and the specified syslog protocol.
Extraction based on specified rules	e_anchor	Extracts strings by using the rules specified by anchor_rules.

e_regex

The e_regex function extracts the value of a field based on a regular expression and assigns the value to other fields.

• Syntax

e_regex(key,Regular expression,fields_info,mode="fill-auto",pack_json=None)

• Parameters

Parameter	Туре	Required	Description
key	Arbitrary	Yes	The name of the input field. If the field that you specify does not exist, no operations are performed. For information about how to specify special field names, see Event types .

Regular expression	String	Yes	The regular expression that is used to extract the value of the field. Regular expressions that contain capturing groups and non-capturing groups are supported. Note Regular expressions that contain non-capturing groups are used in some scenarios. A non-capturing group uses a prefix that consists of a question mark and a colon (2:). Example: \w+e\\v+\.\w(?:\.\cn)?. For more information about non-capturing groups, see Non-capturing group.
fields_info	String/ List/ Dict	No	The names of the fields to which the extracted value is assigned. If you do not specify named capturing groups in the regular expression, you must configure this parameter.
mode	String	No	The overwrite mode of fields. Default value:fill-auto. For information about other values of this parameter, see Field extraction check and overwrite modes.
pack_json	String	No	The field into which the fields specified by fields_info are packed. Default value: None. This value indicates that no fields are packed.

Response

A log that contains new fields is returned.

Examples

 $\circ~$ Example 1: Extract a value that matches the specified regular expression from a field.

Raw log

msg: 192.168.0.1 http://... 127.0.0.0

Transformation rule

Extract the first IP address from the msg field. e_regex("msg",r"\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}","ip")

Result

msg: 192.168.0.1 http://... 127.0.0.0
ip: 192.168.0.1

$\circ~$ Example 2: Extract multiple values that match the specified regular expression from a field.

Raw log

msg: 192.168.0.1 http://... 127.0.0.0

Transformation rule

Extract two IP addresses from the msg field and assign one IP address to server_ip and the other IP address to client_ip. e_regex("msg",r"\d{1,3}\.\d

Result

msg: 192.168.0.1 http://... 127.0.0.0
server_ip: 192.168.0.1
client_ip: 127.0.0.0

• Example 3: Use a capturing group to extract multiple values that match the specified regular expression from a field.

Raw log

content: start sys version: deficience, err: 2

Transformation rule

Extract the values for version and error from the content field by including a capturing group in the regular expression. e_regex("content",r"start sys version: (\w+),\s*err: (\d+)",["version","error"])

Result

content: start sys version: deficience, err: 2 error: 2 version: deficience

 $\circ~$ Example 4: Use a named capturing group to extract multiple values from a field.

Raw log

content: start sys version: deficience, err: 2

Transformation rule

e_regex("content",r"start sys version: (?P<version>\w+),\s*err: (?P<error>\d+)")

Result

content: start sys version: deficience, err: 2 error: 2 version: deficience • Example 5: Use a capturing group in the specified regular expression to extract the value of the dict field and dynamically generate a field name for the value and reformat the value.

Raw log

dict: verify:123

Transformation rule

e_regex("dict",r"(\w+):(\d+)",{r"k_\1": r"v_\2"})

Result

dict: verify:123 k verify: v 123

- Example 6: Extract a value that matches the specified regular expression from a field and pack the result into the name field.
 - Raw log

msg: 192.168.0.1 http://... 127.0.0.0

Transformation rule

e_regex("msg", r"\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}", "ip", pack_json="name")

Result

msg:192.168.0.1 http://... 127.0.0.0 name:{"ip": "192.168.0.1"}

- Example 7: Use the specified regular expression to extract the value of the dict field, dynamically generate a field name for the value and reformat the value, and then pack the result into the name field.
 - Raw log

dict: x:123, y:456, z:789

Transformation rule

e_regex("dict", r"(\w+):(\d+)", {r"k_\1": r"v_\2"}, pack_json="name")

Result

dict:x:123, y:456, z:789 name:{"k_x": "v_123", "k_y": "v_456", "k_z": "v_789"}

- Example 8: Use a capturing group to extract multiple values that match the specified regular expression and pack the result into the name field.
- Raw log

content: start sys version: deficience, err: 2

Transformation rule

e_regex("content", r"start sys version: (\w+),\s*err: (\d+)", ["version", "error"],pack_json="name")

Result

content:start sys version: deficience, err: 2 name:{"version": "deficience", "error": "2"}

e json

The e_json function performs operations on JSON objects in a specified field. You can configure the parameters to expand JSON data, extract JSON data by using the JMESPath expression, or expand the extracted JSON data.

Svntax

```
e_json(key, expand=None, depth=100, prefix="__", suffix="__", fmt="simple", sep=".",
     expand_array=True, fmt_array="{parent}_{index}",
     include_node=r"[\u4e00-\u9fa5\u0800-\u4e00a-zA-Z][\w\-\.]*",
     exclude_node="", include_path="", exclude_path="",
     jmes="", output="", jmes_ignore_none=False, mode='fill-auto'
)
```

② Note If you use the e_json function to parse a string that does not follow the JSON syntax, the function does not parse the string and returns the original string.

Parameters

Parameter	Туре	Required	Description
key	String	Yes	The name of the input field. If the field that you specify does not exist, no operations are performed. For information about how to specify special field names, see Event types.
expand	Boolean	No	 Specifies whether to expand the input field. If you do not configure the imes parameter, the value True is used for the expand parameter by default. The value True indicates that the input field is expanded. If you configure the imes parameter, the value False is used for the expand parameter by default. The value False indicates that the input field is not expanded.

depth	Number	No	The depth to which the function expands the input field. Valid values: 1 to 2000. Default value: 100. The value 1 indicates that only the first level of the field is expanded.
prefix	String	No	The prefix that you want to add to an expanded field.
suffix	String	No	The suffix that you want to add to an expanded field.
			 The formatting method of an expanded field. Valid values: simple (default): The name of the current node is used as the field name. The format is {prefix}{current}{suffix}. full: The name of the current node and the names of all parent nodes are combined and used as the field name. The format is {parent_list_str}{sep}{prefix}(current) {suffix}. The delimiter is specified by the sep parameter. The default delimiter is a period (.).
fmt	String	No	 parent: The name of the current node and the name of the nearest parent node are combined and used as the field name. The format is (parent)(sep[{prefix}(current)(suffix)]. The delimiter is specified by the sep parameter. The default delimiter is a period (.). root: The name of the current node and the name of the root node are combined and used as the field name. The format is (parent_list[0])(sep)(prefix)(current)(suffix)]. The delimiter is specified by the sep parameter. The delimiter is a period (.).
sep	String	No	The delimiter that is used to separate parent and child nodes when the function formats data. If you set the fmt parameter to full, parent, or root, you must configure this parameter. Default value:
expand_array	Boolean	No	Specifies whether to expand the input field into an array. Default value: True . This value indicates that the input field is expanded into an array.
fmt_array	String	No	The formatting method that is used to expand the input field into an array. The format is {parent_rlist[0]}_{index} . You can also use up to five of the following placeholders to expand the input field: parent_list , current , sep , prefix , and suffix .
include_node	String/ Number	No	The whitelist of node names based on which filtering is performed. By default, node names that contain digits, letters, underscores (_), periods (.), and hyphens (-) are automatically expanded.
exclude_node	String	No	The blacklist of node names based on which filtering is performed.
include_path	String	No	The whitelist of node paths based on which filtering is performed.
exclude_path	String	No	The blacklist of node paths based on which filtering is performed.
jmes	String	No	The JMESPath expression that is used to convert field values to JSON objects and extract a specific value.
output	String	No	The field name that is returned for the value extracted by using the JMESPath expression.
jmes_ignore_none	Boolean	No	Specifies whether to skip a field if the value of the field cannot be extracted by using the JMESPath expression. Default value: True. This value indicates that a field is skipped if the value of the field cannot be extracted by using the JMESPath expression. If you specify False for the jmes_ignore_none parameter, an empty string is returned in the same situation.
mode	String	No	The overwrite mode of fields. Default value:fill-auto. For information about other values of this parameter, see Field extraction check and overwrite modes.

• JSON field expanding and filtering

If a whitelist of node names is specified, only the node names included in the whitelist are returned. For example, e_json("json_data_filed", ..., include_node=r'key\d+') specifies a whitelist of node names in the regular expression.

If a blacklist of node names is specified, only the node names included in the blacklist are not returned. For example, e_json("json_data_filed", ..., exclude_node=r'key\d+') specifies a blacklist of node names in the regular expression.

• The regular expressions include_path and exclue_path are used to match node paths from the beginning. Periods (.) are used to separate the paths that match the regular expressions.

• JMESPath-based filtering

JMESPath expressions are used to select and compute data.

- Select a list of element attributes from a specified JSON path: e_json(..., jmes="cve.vendors[*].product",output="product") .
- Concatenate element attributes from a specified JSON path by using commas (,): e_json(..., jmes="join(',', cve.vendors[*].name)", output="vendors")
- Calculate the maximum value of each attribute for each element in a specified JSON path: e_json(..., jmes="max(words[*].score)", output="hot_word")

• Return an empty string if a specified JSON path does not exist or is empty: e_json(..., jmes="max(words[*].score)", output="hot_word", jmes_ignore_none=False)

parent_list and parent_rlist

The following examples show how to use parent_list and parent_rlist: Raw log

Raw lug

data: { "k1": 100,"k2": {"k3": 200,"k4": {"k5": 300}}}

parent_list sorts the parent nodes from left to right.

e_json("data", fmt='{parent_list[0]}-{parent_list[1]}#{current}')

Result:

data:{ "k1": 100,"k2": {"k3": 200,"k4": {"k5": 300}}}
data-k2#k3:200
data-k2#k5:300

parent_rlist sorts the parent nodes from right to left.

e_json("data", fmt='{parent_rlist[0]}-{parent_rlist[1]}#{current}')

Result:

data:{ "k1": 100,"k2": {"k3": 200,"k4": {"k5": 300}}}
k2-data#k3:200
k4-k2#k5:300

Response

A log that contains new fields is returned.

• Examples

Example 1: Expand a field.

Raw log

data: {"k1": 100, "k2": 200}

Transformation rule

e_json("data",depth=1)

Result

data: {"k1": 100, "k2": 200}
k1: 100

```
k2: 200
```

• Example 2: Add a prefix and a suffix to an expanded field.

Raw log

data: {"k1": 100, "k2": 200}

Transformation rule

e_json("data", prefix="data_", suffix="_end")

Result

data: {"k1": 100, "k2": 200}
data_k1_end: 100

data_k2_end: 200

$\circ~$ Example 3: Expand a field in different formats.

Raw log

data: {"k1": 100, "k2": {"k3": 200, "k4": {"k5": 300} } }

Expand a field in the full format.

e_json("data", fmt='full')
data: {"k1": 100, "k2": {"k3": 200, "k4": {"k5": 300} } }
data.k1: 100
data.k2.k3: 200
data.k2.k4.k5: 300

Expand a field in the parent format.

e_json("data", fmt='parent')

data: {"k1": 100, "k2": {"k3": 200, "k4": {"k5": 300} } }
data.k1: 100
k2.k3: 200
k4.k5: 3000

Expand a field in the root format.

e_json("data", fmt='root')

data: {"k1": 100, "k2": {"k3": 200, "k4": {"k5": 300} } }
data.k1: 100
data.k3: 200
data.k5: 300

• Example 4: Configure the sep parameter, prefix parameter, and suffix parameter to extract JSON data.

Raw log

data: {"k1": 100, "k2": {"k3": 200, "k4": {"k5": 300} } }

Transformation rule

e_json("data", fmt='parent', sep="@", prefix="__", suffix="__")

Result

data: {"k1": 100, "k2": {"k3": 200, "k4": {"k5": 300} })
data@_k1__: 100
k2@_k3__: 200
k4@_k5 : 300

• Example 5: Configure the fmt_array parameter to extract JSON data as an array.

Raw log

people: [{"name": "xm", "sex": "boy"}, {"name": "xz", "sex": "boy"}, {"name": "xt", "sex": "girl"}]

Transformation rule

e_json("people", fmt='parent', fmt_array="{parent_rlist[0]}-{index}")

Result

people: [{"name": "xm", "sex": "boy"}, {"name": "xz", "sex": "boy"}, {"name": "xt", "sex": "girl"}]
people-0.name: xm
people-0.sex: boy
people-1.name: xz
people-1.sex: boy

people-2.name: xt
people-2.sex: girl

• Example 6: Extract a JSON object by using the JMESPath expression.

Raw log

data: { "people": [{"first": "James", "last": "d"}, {"first": "Jacob", "last": "e"}], "foo": {"bar": "baz"}}

Transformation rule

e_json("data", jmes='foo', output='jmes_output0')
e_json("data", jmes='foo.bar', output='jmes_output1')
e_json("data", jmes='people[0].last', output='jmes_output2')
e_json("data", jmes='people[*].first', output='jmes_output3')

Result

```
data: { "people": [{"first": "James", "last": "d"}, {"first": "Jacob", "last": "e"}], "foo": {"bar": "baz"})
jmes_output0: {"bar": "baz"}
jmes_output1: baz
jmes_output2: d
jmes_output3: ["james", "jacob"]
```

e_csv, e_psv, and e_tsv

The e_csv function, e_psv function, and e_tsv function extract multiple fields from a specified input field by using a specified delimiter and predefined field names.

- e_csv: uses a comma (,) as the default delimiter.
- e_psv: uses a vertical bar (|) as the default delimiter.
- e_tsv: uses a tab (\tabla) as the default delimiter.

Syntax

```
e_csv(Input field name, Output field list, sep=",", quote='"', restrict=True, mode="fill-auto")
e_psv(Input field name, Output field list, sep="|", quote='"', restrict=True, mode="fill-auto")
e_tsv(Input field name, Output field list, sep="\t", quote='"', restrict=True, mode="fill-auto")
```

Parameters

Parameter	Туре	Required	Description
Input field name	Arbitrary	Yes	The name of the input field. If the field that you specify does not exist, no operations are performed. For information about how to specify special field names, see Event types.
Output field list	Arbitrary	Yes	The names of fields that are returned after the value of the input field is separated by using the specified delimiter. The field names can be in a string list. Example: ["error", "message", "result"]. If the field names do not contain commas (,), you can use commas (.) as delimiters to separate the string. Example: "error, message, result". For information about how to specify special field names, see Event types.
sep	String	No	The delimiter that is used to separate the value of the input field. You must specify a single character as a delimiter.

quote	String	No	The quote that is used to enclose a value. If a value contains a delimiter, you must configure this parameter.
restrict	Boolean	No	 Specifies whether to enable the restricted mode. Default value: False, which indicates that the restricted mode is disabled. If the number of values that are separated with the delimiter in the value of the input field differs from the number of output field names, the operation that is performed by the function varies based on the mode. If the restricted mode is enabled, the function does not perform operations. If the restricted mode is disabled, the function matches the specified fields to the values and assigns specific values to the fields.
mode	String	No	The overwrite mode of fields. Default value:fill-auto. For information about other values of this parameter, see Field extraction check and overwrite modes.

Response

A log that contains new fields is returned.

• Example

In this example, the e_csv function is used. The e_psv function and e_tsv function work in a similar manner to the e_csv function.

• Raw log

content: 192.168.0.100,10/Jun/2019:11:32:16 +0800,example.aliyundoc.com,GET /zf/11874.html
HTTP/1.1,200,0.077,6404,192.168.0.100:8001,200,0.060,https://image.developer.aliyundoc.com/s?
q=%E8%98%88%88%8%%B1%E9%BE%99%89%81%BB%E9%90%2%E7%9A%84%E5%81%9A%E6%B3%95&from=wy878378&uc_param_str=dnntnwvepffrgibijbprsvdsei,,Mozilla/5.0 (Linux; Android 9; HWI-AL00 Build/HUAWEIHWI-AL00) AppleWebKit/537.36,-,-

• Transformation rule

e_csv("content", "remote_addr, time_local,host,request,status,request_time,body_bytes_sent,upstream_addr,upstream_status, upstream_response_time,http_referer,http_x_forwarded_for,http_user_agent,session_id,guid")

• Result

content: 192.168.0.100,10/Jun/2019:11:32:16 +0800,example.aliyundoc.com,GET /zf/11874.html
HTTP/1.1,200,0.077,6404,192.168.0.100:8001,200,0.060,https://image.developer.aliyundoc.com/s?
q=%E8%9B%8B%E8%8A%B1%E9%BE%99%E9%A1%BB%E9%9D%A2%E7%9A%84%E5%81%9A%E6%B3%95&from=wy878378&uc_param_str=dnntnwvepffrgibijbprsvdsei,-
,Mozilla/5.0 (Linux; Android 9; HWI-AL00 Build/HUAWEIHWI-AL00) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Mobile Safari/537.36,-,-
body_bytes_sent: 6404
guid: -
host: example.aliyundoc.com
http_referer: https://image.developer.aliyundoc.com/s?
$q = \$ 88 \$98 \$88 \$88 \$81 \$ 29 \$ B 29 \$ B 29 \$1 \$ B 29 \$90 \$ A 2 \$ 27 \$94 \$ 4 \$ 5 \$ 81 \$9A \$ E 6 \$ B 3 \$ 95 \$ from = wy 87 8378 \$ uc_param_str=dnntnwvepffrgibijbprsvdsei$
http_user_agent: Mozilla/5.0 (Linux; Android 9; HWI-ALOO Build/HUAWEIHWI-ALOO) AppleWebKit/537.36
http_x_forwarded_for: -
remote_addr: 192.168.0.100
request: GET /zf/11874.html HTTP/1.1
request_time: 0.077
session_id: -
status: 200
time_local: 10/Jun/2019:11:32:16 +0800
topic: syslog-forwarder
upstream_addr: 192.168.0.100:8001
upstream_response_time: 0.060
upstream status: 200

e_kv

The e_kv function extracts key-value pairs from multiple input fields by using a specified quote.

Syntax

e_kv(Input field name or input field list, sep="=", quote='"', escape=False, prefix="", suffix="", mode="fill-auto")

Parameters

Parameter	Туре	Required	Description
Input field name or input field list	String or string list	Yes	The name of the input field or the names of multiple input fields. For information about how to specify special field names, see Event types .
sep	String	No	The delimiter that is used to separate a key and the value of the key in a regular expression. Default value: - You can specify one or more characters as a delimiter. Note You can use non-capturing groups in a regular expression, but you cannot use capturing groups in a regular expression. For more information about grouping, see Group.

quote	String	No	The quote that is used to enclose a value. Default value: " . Note We recommend that you configure the quote parameter to enclose a value extracted from a dynamic key-value pair. Examples: a="abc" and b="xyz". If you do not configure the quote parameter, the extracted values can contain only the following characters: letters, digits, underscores (_), hyphens (-), periods (.), percent signs (%), and tildes (~). For example, you can extract a: abl2%~ and b: 123 from a=abl2%~ abc b=l23.
escape	Boolean	No	Specifies whether to extract escape characters in the value of the input field. Default value: False . This value indicates that escape characters in the value of the input field are not extracted. For example, the value <code>abc\</code> of the <code>key</code> field is extracted from the expression <code>key="abc\"xyz"</code> by default. If the <code>escape</code> parameter is set to True, the extracted value is <code>abc"xyz</code> .
prefix	String	No	The prefix that is added to an extracted field.
suffix	String	No	The suffix that is added to an extracted field.
mode	String	No	The overwrite mode of fields. Default value:fill-auto. For information about other values of this parameter, see Field extraction check and overwrite modes.

Response

A log that contains new fields is returned.

• Examples

• Example 1: Extract key-value pairs by using the default delimiter =.

Raw log

http_refer: https://video.developer.aliyundoc.com/s?q=asd&a=1&b=2

Transformation rule

e_kv("http_refer")

Result

http_refer: https://video.developer.aliyundoc.com/s?q=asd&a=1&b=2
q: asd
a: 1

b: 2

$\circ~$ Example 2: Add a prefix and a suffix to extracted fields.

Raw log

http_refer: https://video.developer.aliyundoc.com/s?q=asd&a=1&b=2

Transformation rule

e_kv(
 "http_refer",
 sep="=",
 quote=""',
 escape=False,
 prefix="data_",
 suffix="_end",
 mode="fill-auto",

) • Result

```
http_refer: https://video.developer.aliyundoc.com/s?q=asd&a=1&b=2
data_q_end: asd
data_a_end: 1
data_b_end: 2
```

- Example 3: Extract key-value pairs from the content2 field and extract escape characters by using the escape parameter.
- Raw log

content2: k1:"v1\"abc", k2:"v2", k3: "v3"

Transformation rule

e_kv("content2", sep=":", escape=True)

```
    Result
```

```
content2: k1:"v1\"abc", k2:"v2", k3: "v3"
k1: v1"abc
k2: v2
k3: v3
```

e_kv_delimit

The e_kv_delimit function extracts key-value pairs from input fields by using a specified delimiter.

• Syntax

e_kv_delimit(Input field name or input field list, pair_sep=r"\s", kv_sep="=", prefix="", suffix="", mode="fill-auto")

Parameters

Parameter	Туре	Required	Description
Input field name or input field list	String or string list	Yes	The name of the input field or the names of multiple input fields. For information about how to specify special field names, see Event types.
pair_sep	String	No	The regular expression that is used to separate key-value pairs. Default value: \s . You can also specify $\s \w$ or abc\s . One of the set of the se
kv_sep	String	No	The regular expression that is used to separate key-value pairs. Default value: - The regular expression can contain one or more characters. Note You can use non-capturing groups in a regular expression, but you cannot use capturing groups in a regular expression. For more information about grouping, see Group.
prefix	String	No	The prefix that is added to an extracted field.
suffix	String	No	The suffix that is added to an extracted field.
mode	String	No	The overwrite mode of fields. Default value:fill-auto. For information about other values of this parameter, see Field extraction check and overwrite modes.

Response

A log that contains new fields is returned.

• Examples

- Example 1: Extract key-value pairs by using the default delimiter = .
 - Raw log

data: i=c1 k1=v1 k2=v2 k3=v3

() Note If the raw log is request_uri: a1=16a2=6a3=3 and the value of a2 is empty, the e_kv_delimit() function cannot extract the value of a2. You can use the e_regex() function to extract the value of a2. Example: e_regex("request_uri",r'(\w+)=([^=&]*)', {r"\1":r"\2"}, mode="overwrite").

Transformation rule

e_kv_delimit("data")

```
    Result
```

data: i=c1 k1=v1 k2=v2 k3=v3 i: c1 k2: v2 k1: v1

k3: v3

- Example 2: Extract key-value pairs by using the delimiters 6?
- Raw log

data: k1=v1&k2=v2?k3=v3

Transformation rule

e_kv_delimit("data",pair_sep=r"&?")

Result

```
data: k1=v1&k2=v2?k3=v3
k2: v2
k1: v1
k3: v3
```

- $\circ~$ Example 3: Extract key-value pairs by using a regular expression.
 - Raw log

data: k1=v1 k2:v2 k3=v3

Transformation rule

e_kv_delimit("data", kv_sep=r"(?:=|:)")

Result

data: k1=v1 k2:v2 k3=v3
k2: v2
k1: v1
k3: v3

e_syslogrfc

The e_syslogrfc function calculates the values of the facility and severity fields and returns the value of the facilitylabel field that indicates level information. The function calculates the values based on the value of the priority field and the specified syslog protocol.

• Syntax

e_syslogrfc(key, rfc, fields_info=None, mode='overwrite')

Parameters

Parameter	Туре	Required	Description
key	Arbitrary	Yes	The name of the input field. You must enter a field that indicates a $\ensuremath{\mbox{\sc priority}}$.
rfc	String	Yes	The syslog protocol that is used. The syslog protocols are defined in RFC. Valid values: SYSLOGRFC3164 and SYSLOGRFC5424.
fields_info	Dict	No	<pre>key indicates the name of the input field, and value indicates the name of the new field. The following fields can be renamed. The new names can be modified. ("_severity_":"sev", "_facility_":"fac","_severitylabel_ ":"sevlabel", "_facilitylabel_":"faclabel"}</pre>
mode	String	No	The overwrite mode of fields. Default value:overwrite. For information about other values of this parameter, see Field extraction check and overwrite modes.

Response

A log that contains new fields is returned.

Examples

- Example 1: Extract the values of the facility field and severity field and return level information based on the syslog protocol defined in RFC 5424.
- Raw log

receive_time: 1558663265
priority: 13
version: 1
_log_time_: 2019-05-06 11:50:16.015554+08:00
hostname: 12bp1a65******i2qZ
program: root
procid: _msgid_: _extradata_: _content_: twish

Transformation rule

e_syslogrfc("_priority_","SYSLOGRFC5424")

Result

```
receive_time: 1558663265
_priority_: 13
_version_: 1
_log_time_: 2019-05-06 11:50:16.015554+08:00
_hostname_: i2bpla65******i2qZ
_program_: root
_procid_: -
_msgid_: -
_extradata_: -
_content_: twish
_facility_: 1
_severity_label_: Notice: normal but significant condition
_facilitylabel_: user-level messages
```

• Example 2: Extract the values of the facility field and severity field and return level information based on the syslog protocol defined in RFC 5424. Then, rename the fields by configuring the fields_info parameter.

Raw log

```
receive_time: 1558663265
_priority_: 13
_version_: 1
_log_time_: 2019-05-06 11:50:16.015554+08:00
_hostname_: 1Zbpla65*******12qZ
_program_: root
_procid_: -
_msgid_: -
_extradata_: -
_content_: twish
```

Transformation rule

```
e_syslogrfc(
    "_priority_",
    "SYSLOGRFC5424",
    {
        "_facility_": "fac",
        "_severity_": "sev",
        "_facilitylabel_": "_facility_label_",
        "_severitylabel_": "_severity_label_",
    },
)
```

Result

```
receive_time: 1558663265
_priority_: 13
_version_: 1
log_time_: 2019-05-06 11:50:16.015554+08:00
_hostname_: iZbpla65*******i2qZ
_program_: root
_procid_: -
_extradata_: -
_content_: twish
_facility_: 1
_severity_: 5
_severity_: 5
_severity_izbel_: Notice: normal but significant condition
_facility_label_: user-level messages
```

e_anchor

The e_anchor function extracts strings by using the rules specified by anchor_rules.

Syntax

e_anchor(key,anchor_rules,fields,restrict=False,mode="overwrite")

Parameters

Cloud Defined Storage

Parameter	Туре	Required	Description
key	Arbitrary	Yes	The name of the field.
anchor_rules	String	Yes	The rules that are used to extract strings. Examples: User = *; Severity = *;, Asterisks (*) indicate the content that you want to extract. By default, a space is specified before Value in the logs that are displayed in the Key : Value format in the Log Service console. When you configure the anchor_rules parameter, remove the default space.
fields	Arbitrary	Yes	The names of the output fields whose values are extracted from the value of input field. The field names can be in a string list. Example: ["user", "job", "result"] . If the field names do not contain commas (.), you can use commas (.) to separate the string. Example: "user, job, result" . For information about how to specify special field names, see Event types. Special field names can contain special characters except asterisks (*). You can use an asterisk (*) to skip a field. For example, only user and result are extracted from "user, *, result" . For more information, see Example 10.
restrict	Boolean	No	 Specifies whether to enable the restricted mode. Default value: False. This value indicates that the restricted mode is disabled. If the number of values that are extracted from the value of the operation that is performed by the function varies based on the mode. If the restricted mode is enabled, the function does not perform operations. If the restricted mode is disabled, the function matches the specified fields to the values and assigns specific values to the fields.
mode	String	No	Default value: overwrite. For more information, see Field extraction check and overwrite modes.

Response

The extracted data is returned.

- Examples
- $\circ~$ Example 1: Extract the values for specified fields from a log.
 - Raw log

content : "Aug 2 04:06:08: host=192.168.0.10: local/ssl2 notice mcpd[3772]: User=jsmith@example.com: severity=warning: 01070638:5: Pool member 172.31.51.22:0 monitor status down."

Transformation rule

e_anchor("content", "User=*: severity=*:", ["user_field", "severity_field"])

Result

content : "Aug 2 04:06:08: host=192.168.0.10: local/ssl2 notice mcpd[3772]: User=jsmith@example.com: severity=warning: 01070638:5: Pool
member 172.31.51.22:0 monitor status down."
user_field : jsmith@example.com
severity_field : warning

• Example 2: Extract multiple values in the JSON array format.

Raw log

content : '"information":{"name_list":["Twiss","Evan","Wind","like"],"university":["UCL","Stanford
University","CMU"]},"other":"graduate"'

Transformation rule

e_anchor("content", 'name_list":*, "university":*}, ', ["name_list", "universities"])

Result

content : '"information":{"name_list":["Twiss","Evan","Wind","like"],"university":["UCL","Stanford University","CMU"]},"other":"graduate"' name_list : ["Twiss","Evan","Wind","like"] universities : ["UCL","Stanford University","CMU"] · Example 3: Extract a log that contains special characters

Raw log

content : (+2019) June 24 "I am iron man"

Transformation rule

e_anchor("content", "(+*) * \"*\"",["Year","Date","Msg"])

Result

content : (+2019) June 24 "I am iron man" Year : 2019 Date : June 24 Msg : I am iron man

• Example 4: Extract a log that contains the control character \x09.

Raw log

content : \x09\x09\x09Chrome/55.0 Safari/537.36

Transformation rule

e_anchor("content", "\x09\x09\x09*/55.0 */537.36",["Google", "Apple"])

Result

content : \x09\x09\x09Chrome/55.0 Safari/537.36 Google : Chrome Apple : Safari

- Example 5: Extract the field content that contains special characters. To...Subject that comes after MESSAGE: is the actual content of the content field.
 - Raw log

content : 12:08:10,651 INFO sample_server ReportEmailer:178 - DEBUG SENDING MESSAGE: To: example@aliyun.com Subject: New line Breaks in Message

Transformation rule

e_anchor("content","* INFO *: \n To: *\n Subject: *",["time","message","email","subject"])

Result

content : 12:08:10,651 INFO sample_server ReportEmailer:178 - DEBUG SENDING MESSAGE: To: example@aliyun.com Subject: New line Breaks in Message

time : 12:08:10,651 message : sample_server ReportEmailer:178 - DEBUG SENDING MESSAGE email : example@aliyun.c subject : New line Breaks in Message

• Example 6: Extract the field content that contains special characters and return the value that does not display the control character \t .

Raw log

content : I'm tabbed in

Transformation rule

e_anchor("content","\tI'm * in","word")

You can also use the following transformation rule to copy the value of the content field. Remove the default space from the value. e_anchor("content"," I'm * in","word")

Result

content : I'm tabbed in word : tabbed

• Example 7: Extract the field content that contains special characters and return the value that displays the control character 🗽

Raw log

content : \tI'm tabbed in

- Transformation rule
 - e_anchor("content","\tI'm * in","word")
 - # You can also use the following transformation rule: e_anchor("content"," I'm * in","word")
- Result

content : \tI'm tabbed in word : tabbed

• Example 8: Extract logs in restricted mode.

Raw log

content : I used to love having snowball fight with my friends and building snowmen on the streets around our neighborhood

Transformation rule

e_anchor("content","I * to * having",["v_word", "n_word","asd"],restrict=True)

Result

content : I used to love having snowball fight with my friends and building snowmen on the streets around our neighborhood

• Example 9: Extract logs in non-restricted mode.

Raw log

content : I used to love having snowball fight with my friends and building snowmen on the streets around our neighborhood

Transformation rule

e_anchor("content","love * fight with my * and",["test1","test2","test13"],restrict=False)

```
    Result
```

content : I used to love having snowball fight with my friends and building snowmen on the streets around our neighborhood
test1 : having snowball
test2 : friends

- Example 10: Extract the value of a field and assign the extracted value to another field.
- Raw log

content: Could you compare the severity of natural disasters to man-made disasters

Transformation rule

e_anchor('content', 'compare the * of natural disasters to man-made *', 'n-word,*')

Result

content : Could you compare the severity of natural disasters to man-made disasters n-word : severity

4.5.8.6.6. Mapping and enrichment functions

This topic describes the syntax and parameters of mapping and enrichment functions. This topic also provides examples on how to use the functions.

Functions

Category	Function	Description
Field based mapping	e_dict_map	Maps the value of an input field to a value in a specified dictionary and returns a new field.
neu-based mapping	e_table_map	Maps the value of an input field to a row in a specified table and returns a new field.
Search based mapping	e_search_dict_map	Searches the keywords in a specified dictionary for a raw log field, maps the field to a value in the dictionary, and returns a new field. The keywords must be query strings.
Search-based mapping	e_search_table_map	Searches a specified column in a specified table for a raw log field, maps the field to a row in the table, and returns a new field. The values of the column must be query strings.

e_dict_map

The e_dict_map function maps the value of an input field to a value in a specified dictionary and returns a new field.

Syntax

e dict map(data, field, output field, case insensitive=True, missing=None, mode="overwrite")

Parameters

Parameter	Туре	Required	Description
data	Dict	Yes	The dictionary that is used for mapping. The value of this parameter must be in the {key01:value01,key01:value02,} standard format. The keys must be strings. Example: {"1": "TCP", "2": "UDP", "3": "HTTP", "*": "Unknown"}.
field	String or string list	Yes	 One or more field names. If the value of this parameter contains multiple field names, the system performs the following operations: The system performs mapping on the field names in sequence. If the system matches multiple values for the fields and the mode parameter is set to overwrite, the system returns the value that is last matched. If the system matches no values for the fields, the system returns the value of the missing parameter.

output_field	String	Yes	The name of the output field.
case_insensitive	Boolean	No	 Specifies whether data is considered not case-sensitive when the system performs mapping. Default value: True. This value indicates that data is considered not case-sensitive. Note If the dictionary contains a key for which the letter cases are different and the case insensitive parameter is set to True, the system first maps the value of the input field to the key that uses the same case as the value. If the key does not exist, the system randomly maps the value to one of the multiple keys.
missing	String	No	The value that is assigned to the field specified by output_field when no match is found for the input field. Default value: None. This value indicates that no assignment is performed. Note If the dictionary contains a key of an asterisk (*), the missing parameter becomes invalid. This is because an asterisk (*) has a higher priority than the missing parameter.
mode	String	No	The overwrite mode of fields. Default value:overwrite. For more information, see Field extraction check and overwrite modes

- Response
 - A log that contains a new field is returned.
- Examples
 - Example 1: Map the value of the pro field to a value in a dictionary and generate a new field named protocol.

•	Raw log	9
	data:	123
	pro:	1

Transformation rule

```
e_dict_map(
    {"1": "TCP", "2": "UDP", "3": "HTTP", "6": "HTTPS", "*": "Unknown"},
    "pro",
    "protocol",
)
```

```
    Result
```

```
data: 123
pro: 1
protocol: TCP
```

- Example 2: Map the values of the **status** field to values in a dictionary and generate a new field named **message**.
 - Raw logs

status:	500
status:	400
status:	200

Transformation rule

e_dict_map({"400": "Error", "200": "Success", "*": "Other"}, "status", "message")

Result

status:	500
message:	Other
status:	400
message:	Error
status:	200
message:	Success

e_table_map

The e_table_map function maps the value of an input field to a row in a specified table and returns a new field.

Syntax

e_table_map(data, field, output_fields, missing=None, mode="fill-auto")

• Parameters

Parameter	Туре	Required	Description
-----------	------	----------	-------------

data	Table	Yes	The table that is used for mapping. Image: The table that is used for mapping. Image: The table t
field	String, string list, or tuple list	Yes	The input field. If a log does not contain the field, no operations are performed on the log.
output_fields	String, string list, or tuple list	Yes	The output fields. Example: ["province", "pop"] .
missing	String	No	The value that is assigned to the field specified by output field when no match is found for the input field. Default value: None. This value indicates that no assignment is performed. If you want to map the input field to multiple columns, you can set the missing parameter to a list of default values that correspond to the input field. The number of the default values must be the same as the number of the columns. (? Note If the table contains a column of an asterisk (*), the missing parameter becomes invalid. This is because an asterisk (*) has a higher priority than the missing parameter.
mode	String	No	The overwrite mode of fields. Default value:fill-auto. For more information, see Field extraction check and overwrite modes

Response

A log that contains new fields is returned.

- Examples
- Example 1: Map the value of the city field to a row in a table and return the value of the province field for the row.
 - Raw log
 - data: 123 city: nj
 - Transformation rule
 - e_table_map(
 - tab_parse_csv("city,pop,province\nnj,800,js\nsh,2000,sh"), "city", "province"
 -)
 - Result
 - data: 123 city: nj province: js
- Example 2: Map the value of the city field to a row in a table and return the values of the province and pop fields for the row.
 - Raw log

data: 123 city: nj

Transformation rule

```
e_table_map(
```

```
tab_parse_csv("city,pop,province\nnj,800,js\nsh,2000,sh"),
"city",
["province", "pop"],
```

) • Result

```
data: 123
city: nj
province: js
pop: 800
```

User Guide-Log Service

• Example 3: Use the tab_parse_csv function to build a table, map the value of the city field to a row in the table, and return the values of the province and pop fields for the row.

```
    Raw log
```

- data: 123 city: nj
- Transformation rule

```
e_table_map(
    tab_parse_csv("city#pop#province\nnj#800#js\nsh#2000#sh", sep="#"),
    "city",
    ["province", "pop"],
)
```

Result

```
data: 123
city: nj
province: js
pop: 800
```

- Example 4: Use the tab_parse_csv function to build a table, map the value of the city field to a row in the table, and return the values of the province and pop fields for the row.
 - Raw log

data: 123 city: nj

- Transformation rule
 - e_table_map(
 - tab_parse_csv(
 "city,pop,province\n|nj|,|800|,|js|\n|shang hai|,2000,|SHANG,HAI|", quote="|"
),

```
"city",
["province", "pop"],
)
```

- Result
 - data: 123 city: nj province: js pop: 800
- Example 5: The input field is different from the corresponding field in the table that is used for mapping. Find a row in the table based on the **cty** and **city** fields and return the value of the **province** field for the row.
 - Raw log
 - data: 123 cty: nj
 - Transformation rule
 - e_table_map(
 - tab_parse_csv("city,pop,province\nnj,800,js\nsh,2000,sh"),
 [("cty", "city")],
 - "province",
 -)
 - Result

data: 123 cty: nj province: js

- Example 6: The input field is different from the corresponding field in the table that is used for mapping. Map data and rename the output field.
 - Raw log
 - data: 123 cty: nj
 - Transformation rule
 - e_table_map(
 - tab_parse_csv("city,pop,province\nnj,800,js\nsh,2000,sh"),
 [("cty", "city")],
 [("province", "pro")],
 -)
 - Result

data: 123 cty: nj pro: js

- $\circ~$ Example 7: Map the values of multiple fields to a row in a table.
 - Raw log
 - data: 123 city: nj pop: 800
 - Transformation rule

```
e_table_map(
   tab_parse_csv("city,pop,province\nnj,800,js\nsh,2000,sh"),
   ["city", "pop"],
   "province",
```

) • Result

```
data: 123
city: nj
pop: 800
province: js
```

- Example 8: Map the values of multiple fields to a row in a table. The input fields are different from the corresponding fields in the table that is used for mapping.
 - Raw log
 - data: 123 cty: nj pp: 800
 - Transformation rule
 - e_table_map(
 - tab_parse_csv("city,pop,province\nnj,800,js\nsh,2000,sh"), [("cty", "city"), ("pp", "pop")], "province",
 -)
 - Result
 - data: 123 cty: nj pp: 800 province: js

e_search_dict_map

The e_search_dict_map function searches the keywords in a specified dictionary for a raw log field, maps the field to a value in the dictionary, and returns a new field. The keywords must be query strings.

• Syntax

e_search_dict_map(data, output_field, multi_match=False, multi_join=" ", missing=None, mode="overwrite")

• Parameters

Parameter	Туре	Required	Description
data	Dict	Yes	The dictionary that is used for mapping. The value of this parameter must be in the {key01:value01, key01:value02,} standard format. The keys must be query strings.
output_field	String	Yes	The name of the output field that you want the function to return.
multi_match	Boolean	No	Specifies whether the system can match multiple values for the input field. Default value: False. This value indicates that the system does not match multiple values and returns only the value that is matched for the last value of the input field. You can configure the <code>multi_join</code> parameter to concatenate multiple values that are matched for the input field.
multi_join	String	No	The character that is used to concatenate the multiple values that are matched for the input field. The default value is a space. This parameter takes effect only when the multi_match parameter is set to True.
missing	String	No	The value that is assigned to the field specified by output_field when no match is found for the input field. Default value: None. This value indicates that no assignment is performed. Image: The state of the distribution of the distributic of the distribution of the distributic o
mode	String	No	The overwrite mode of fields. Default value:overwrite. For more information, see Field extraction check and overwrite modes
Response

The value that is matched for the input field is returned.

- Examples
- Example 1: Map data.
- Raw log

data:123 pro:1

Transformation rule

e_search_dict_map ({"pro==1": "TCP", "pro==2": "UDP", "pro==3": "HTTP"}, "protocol")

Result

data:123 pro:1 protocol:TCP

 $\circ~$ Example 2: Map data based on the first character of each field value.

Raw log

status:200,300

Transformation rule

```
e_search_dict_map(
    {
        "status:2??": "ok",
        "status:3?": "redirect",
        "status:42?": "auth",
        "status:2??": "server_error",
    },
    "status_desc",
    multi_match=True,
    multi_join="Test",
}
```

Result

status:200,300
status_desc: okTestredirect

e_search_table_map

The e_search_table_map function searches a specified column in a specified table for a raw log field, maps the field to a row in the table, and returns a new field. The values of the column must be query strings.

Syntax

e_search_table_map(data, inpt, output_fields, multi_match=False, multi_join=" ", missing=None, mode="fill-auto")

• Parameters

Parameter	Туре	Required	Description
data	Table	Yes	The table that is used for mapping. The table must contain a column whose values are query strings.
inpt	String	Yes	The name of the column in which the system searches for data. The field that is indicated by the column is considered the input field.
output_fields	String, string list, or tuple list	Yes	The output fields that you want the function to return. The value of this parameter is a string, string list, or tuple list.
multi_match	Boolean	No	Specifies whether the system can match multiple values for the input field. Default value: False. This value indicates that the system does not match multiple values and returns only the value that is matched for the first value of the input field. You can configure the multi_join parameter to concatenate multiple values that are matched for the input field.
multi_join	String	No	The character that is used to concatenate the multiple values that are matched for the input field. The default value is a space. This parameter takes effect only when the multi_match parameter is set to True.
missing	String	No	The value that is assigned to the field specified by output_field when no match is found for the input field. Default value: None. This value indicates that no assignment is performed. ⑦ Note If the table contains a column of an asterisk (*) , the missing parameter becomes invalid. This is because an asterisk (*) has a higher priority than the missing parameter.
mode	String	No	The overwrite mode of fields. Default value:fill-auto. For more information, see Field extraction check and overwrite modes

Response

The value that is matched for the input field is returned.

• Examples

- Example 1: Map the value of the city field to a row in a table and return the values of the pop and province fields for the row.
- Raw log

data: 123 city: sh

The following table shows the mappings between the pop and province fields. The values in the search column are query strings.

search	рор	province
city==nj	800	js
city==sh	2000	sh

Transformation rule

e_search_table_map(
<pre>tab_parse_csv("search,pop,province\ncity==nj,800,js\ncity==sh,2000,sh"),</pre>
"search",
["pop", "province"],
)

Result

data: 123 city: sh province: sh pop: 2000

• Example 2: Map data in overwrite mode.

Raw log

data: 123 city: nj province:

Transformation rule

e_search_table_map(
 tab_parse_csv("search,pop,province\ncity==nj,800,js\ncity==sh,2000,sh"),
 "search",
 "province",
 mode="overwrite",
)

Result

data: 123 city: nj province: js

$\circ~$ Example 3: Map data by specifying a value for the $\ensuremath{\textit{missing}}$ parameter.

Raw log

data: 123 city: wh province:

Transformation rule

e_search_table_map(
 tab_parse_csv("search,pop,province\ncity==nj,800,\ncity==sh,2000,sh"),
 "search",
 "province",
 missing="Unknown",

) • Result

data: 123 city: wh province: Unknown • Example 4: Map data by setting the multi_match parameter to True.

```
    Raw log
```

```
data: 123
city: nj,sh
province:
```

Transformation rule

```
e_search_table_map(
    tab_parse_csv("search,pop,province\ncity:nj,800,js\ncity:sh,2000,sh"),
    "search",
    "province",
    multi_match=True,
    multi_join=",",
)
```

```
    Result
```

```
data: 123
city: nj,sh
province: js,sh
```

4.5.8.6.7. Value-added content function

This topic describes the syntax and parameters of value-added content function. This topic also provides examples on how to use the function.

Functions

Category	Function	Description
Threat intelligence	e_threat_intelligence	Obtains the threat intelligence for an IP address or a domain name that is specified by a log field and assigns the threat intelligence as a value to a specified field.

e_threat_intelligence

The e_threat_intelligence function obtains the threat intelligence for an IP address or domain name that is specified by a log field and assigns the threat intelligence as a value to a specified field.

- If no threat intelligence is found for the specified IP address or domain name, no data is assigned as a value to the specified field, and your data transformation job is not affected.
- Alibaba Cloud Threat Intelligence provides the threat intelligence of the last 30 days and updates the threat intelligence once a day. If you want to obtain detailed threat intelligence, you can activate Threat Intelligence.
- Syntax

e_threat_intelligence(category, field, output_field=None, mode="overwrite")

• Parameters

Parameter	Туре	Required	Description
category	String	Yes	The type of the threat intelligence. Valid values: • <i>ip</i> : obtains the threat intelligence for an IP address. • <i>domain</i> : obtains the threat intelligence for a domain name.
field	String	Yes	The name of the log field that is used to obtain the threat intelligence.
output_field	String	No	The name of the field to which the threat intelligence is assigned as a value. If you do not configure this parameter, the threat intelligence is assigned as a value to the
mode	String	No	The overwrite mode of fields. Default value: overwrite. For more information, see Field extraction check and overwrite modes

Response

The threat intelligence is returned in the JSON format to the field specified by the *output_field* parameter. The following tables describe the parameters in the threat intelligence.

Parameter	Description
confidence	The confidence level of the threat intelligence. The value is an integer within the range of [0,100]. A larger value indicates a higher confidence level.
severity	The threat level of the threat intelligence. O: no risk 1: low risk 2: medium risk 3: high risk 4: critical risk
family	The malware family. An empty string is returned.
ioc_type	The type of the threat intelligence. Valid value: ipv4. Only IPv4 IP addresses are supported.
ioc_raw	The IP address for which the threat intelligence is obtained.
intel_type	The type of the risk tag. Multiple risk tags are separated by commas (,). • web_attack: an IP address from which a network attack is initiated • tor: an IP address of a Top of Rack (TOR) node • mining: an IP address of a mining program • c2: an IP address of a command and control (C2) server • malicious: an IP address of a malicious download source • exploit: an IP address from which an exploit attack is initiated • webshell: an IP address from which a metwork service scan is initiated • scan: an IP address from which a network service scan is initiated
country	The country to which the IP address belongs.
province	The province to which the IP address belongs.
city	The city to which the IP address belongs.
isp	The telecommunications carrier of the network to which the IP address belongs.

• Threat intelligence for a domain name

Parameter	Description
confidence	The confidence level of the threat intelligence. The value is an integer within the range of [0,100]. A larger value indicates a higher confidence level.
severity	The threat level of the threat intelligence. O: no risk 1: low risk 2: medium risk 3: high risk 4: critical risk
family	The malware family. An empty string is returned.
ioc_type	The type of the threat intelligence. Valid value: domain.
ioc_raw	The domain name for which the threat intelligence is obtained.
intel_type	The type of the risk tag. Multiple risk tags are separated by commas (,). For more information, see Risk tags of domain names.
root_domain	The root domain name to which the domain name belongs.

• Examples

- Example 1: Obtain the threat intelligence for an IP address and assign the threat intelligence as a value to a specified field.
- Raw log

remote addr: 203.0.113.1 method: GET

- Transformation rule
- Obtain the threat intelligence for the IP address specified by the remote_addr field and assign the threat intelligence as a value to the threat info field.

e_threat_intelligence("ip", "remote_addr", output_field="threat_info")

Result

threat_info:{ "confidence": 100. "severity": 4, "family": "", "ioc_raw": "203.0.113.1", "ioc_type": "ipv4", "intel_type": "web", "country": "China", "province": "Zhejiang", "city": "Hangzhou", "isp": "China Telecom" } method:GET remote_addr:203.0.113.1

• Example 2: Obtain the threat intelligence for an IP address and assign the threat intelligence as a value to the default field.

Raw log

remote_addr: 203.0.113.1 method: GET

Transformation rule

Obtain the threat intelligence for the IP address specified by the remote_addr field and assign the threat intelligence as a value to the default field.

e_threat_intelligence("ip", "remote_addr")

Result

__threat_intelligence__:remote_addr:{ "confidence": 100, "severity": 4, "family": "", "ioc_raw": "203.0.113.1", "ioc_type": "ipv4", "intel_type": "web", "country": "China", "province": "Zhejiang", "city": "Hangzhou", "isp": "China Telecom" method:GET

- remote_addr:203.0.113.1
- Example 3: Obtain the threat intelligence for a domain name and assign the threat intelligence as a value to a specified field.

Raw log

domain_name: www.02a470ee85e5c43f27b9c42a3c46a8bb.info

Transformation rule

Obtain the threat intelligence for the domain name specified by the domain_name field and assign the threat intelligence as a value to the _ti_ field

e_threat_intelligence("domain", "domain_name", output_field="_ti_")

Result

domain_name: www.02a470ee85e5c43f27b9c42a3c46a8bb.info

- _ti_: { "confidence": 91.
- "severity": 3,
- "family": "",
- "ioc_raw": "www.02a470ee85e5c43f27b9c42a3c46a8bb.info",
- "ioc_type": "domain",
- "root_domain": "02a470ee85e5c43f27b9c42a3c46a8bb.info", "intel_type": "sinkhole;rat_trojan;js_miner"
- }

Appendix

Table 1. Risk tags of domain names

Risk tag	Description	Risk tag	Description
malware	Malware	botnet	Botnet
spy_trojan	Trojan-spy	trojan	Trojan

User Guide-Log Service

worm	Worm	bank_trojan	Banker trojan
ransomware	Ransomware	adware	Adware
backdoor_trojan	Backdoor trojan	exploit	Exploit
hacktool	Hacking tool	malicious_doc	Malicious document
infected_virus	Infectious virus	bootkit_trojan	Bootkit trojan
trojan_dropper	Trojan dropper	script_trojan	Trojan script
riskware	Riskware	virus	Virus
apt	APT	trojan_downloader	Trojan downloader
rat_trojan	Remote access trojan (RAT)	rat	RAT
hijack	Hijack	ddos_trojan	DDoS trojan
macro_virus	Macro virus	spam_email	Spam
porn	Pornographic website	js_miner	JavaScript mining
rootkit_trojan	Rootkit trojan	compromised_host	Compromised host
private_server	Private server	gamble	Gambling website
c2	C2 server	dnslog_attack	DNSLog attack
miner	Mining	infostealer	Information stealer
malicious_group	Malicious group	malicious	Malicious website
sinkhole	Sinkhole	miner_pool	Mining pool
dga	DGA	None	None

4.5.8.7. Expression functions

4.5.8.7.1. Overview of expression functions

The domain-specific language (DSL) for Log Service is provided to construct expression functions that return specific values. This helps you transform log data.

Category	Function	Description
Event check functions	e_has, e_not_has, e_search, e_match, e_match_any, and e_match_all	Checks whether a field exists or whether a field or the value of a field meets a specified condition.
Operator functions	Some op_* functions	Compares values, evaluates values based on a specified condition or container, or performs general-purpose multi-value operations.
Conversion functions	ct_* functions	Converts data types among numeric values, strings, and Boolean values, or converts numbers between different numeral systems.
Arithmetic functions	math_*, mat_*, and some op_* functions	Performs mathematical calculation or multi-value calculation.
String functions	str_* functions	Processes strings.
Date and time functions	dt_* functions	Converts time values among UNIX timestamps, datetime objects, and datetime strings, changes time zones, and returns the difference between two time values.
Regular expression functions	regex_* functions	Extracts, retrieves, replaces, or splits values based on regular expressions.
Grok function	Grok function	Extracts specific values based on regular expressions.
Structured data functions	json_*, xml_*, and gzip_* functions	Extracts or parses fields.
IP address parsing functions	geo_parse and ip_* functions	Parses IP addresses.
Encoding and decoding functions	<pre>url_*, html_*, md5_*, sha1_*, base64_*, ip2long, long2ip, aes_encrypt, and aes_decrypt functions</pre>	Encodes or decodes data.
Parsing functions	ua_* functions	Parses a User-Agent header.
List functions	Some op_* and lst_* functions	Performs operations on a list, including obtaining or modifying a list.
Dictionary functions	Some op_* and dct_* functions	Performs operations on a dictionary, including obtaining or modifying a dictionary.
Table functions	tab_* functions	Constructs a table from text or constructs a dictionary from a table.
Resource functions	res_* functions	Pulls configuration data and data from an Object Storage Service (OSS) bucket, a Logstore, or a table in an ApsaraDB RDS for MySQL database.

This topic describes the syntax and parameters of event check functions. This topic also provides examples on how to use the functions.

Functions

Category	Function	Description	
Pacie function	e_has	Checks whether a log field exists.	
basic function	e_not_has	Checks whether a log field does not exist.	
	e_search	Searches for an event by using a query syntax that is similar to Lucene.	
	e_match	Checks whether the value of a log field meets the conditions specified in a regular expression.	
Expression function	e_match_any	Checks whether the value of a log field meets the conditions specified in a regular expression. If one or more specified fields match the regular expression, True is returned. Otherwise, False is returned.	
	e_match_all	Checks whether the value of a log field meets the conditions specified in a regular expression. If all specified fields match the regular expression, True is returned. Otherwise, False is returned.	

e_has

The e_has function checks whether a log field exists.

• Syntax

- e_has("key")
- Parameters

Parameter	Туре	Required	Description
key	String	Yes	The name of the log field.

Response

If the specified field exists, True is returned. Otherwise, False is returned.

- Example
 - Check whether a log contains the content field. If the log contains the content field, the log is retained. Otherwise, the log is dropped.
- Raw log

	content: 123
0	Transformation rule
	<pre>e_keep(e_has("content"))</pre>
o	Result
	content: 123
e_r	not_has

The e_not_has function checks whether a log field does not exist.

Syntax

e_not_has("key")

• Parameters

Parameter	Туре	Required	Description
key	String	Yes	The name of the log field.

Response

If the specified field does not exist, True is returned. Otherwise, False is returned.

• Example

Check whether a log contains the content field. If the log does not contain the content field, the log is retained. Otherwise, the log is dropped.

- Raw log
- content: 123
- Transformation rule

e_if_else(e_not_has("content"),KEEP,DROP)

- Result
- The log is dropped.

e_search

The e_search function searches for an event by using a query syntax that is similar to Lucene.

Syntax

	e_search(querystring)			
•	Parameters			
	Parameter	Туре	Required	Description

querystring	String	Yes	The query string that you want to use to filter log data. For more information, see Query string syntax.
ResponseIf the specified conditionsExample	are met, True is returned. Othe	rwise, False is returned.	
<pre># Full-text search e_search("active error" operator OR. e_search('"active error</pre>) # Search for multiple subst	trings in full text. The subst	rings are associated with each other by using the logical
<pre># Field search e_search("status: activ e_search('author: "john e_search('field: active e query string in this</pre>	e") # Search for a substring smith"') # Searches for a su error') # Search the specifi example is equivalent to field:	in a specified field. ubstring that contains a space ied field for the substring "a cactive OR "error".	e character in a specified field. Active" or searches all logs for the substring "error". Th
<pre># Exact match e_search('author== "joh</pre>	n smith"')		
<pre># Search for field valu mark (?) to match one c</pre>	es by using wildcard characters haracter.	s. You can use an asterisk (*)	to match zero or more characters. You can use a question
<pre>e_search("status: activ ""). e_search("status: activ</pre>	e*test") # active*test conta e?good") # active?good conta	ains one asterisk (*). You do ains one question mark (?). Yo	not need to enclose the value in double quotation marks (ou do not need to enclose the value in double quotation ma
<pre>rks (""). e_search("status== ac*t</pre>	ive?good") # The query string	is used for exact match.	
<pre># Escape special charac value by using backslas e_search('status: "*\? ""). The asterisks (*), e_search("status: activ ""). e_search("status: activ compared by the search of the search of the search "").</pre>	<pre>ters in a field value. Asterisk hes (\). ()[]:="') # *\?()[]:= contai question marks (?), and backsl e*test") # active*test conta e* test") # active*test contai </pre>	<pre>ks (*) or question marks (?) t ins multiple special character lashes (\) in the value are es nins one asterisk (*). You do ontains one question mark (?).</pre>	that are not used as wildcards must be escaped in a field (s. You must enclose the value in double quotation marks (scaped. not need to enclose the value in double quotation marks (. You do not need to enclose the value in double quotation
<pre># Escape special charac e_search("*\(1+1\))?: racters by using backsl e_search("_tag_\:_co e_search("field name in</pre>	ters in a field name abc")	annot enclose the field name i Ist escape special characters # Enter the Chinese char	n double quotation marks (""). You must escape special cha by using backslashes (\). acters that comprise the field name.
<pre># Search for strings by e_search('content~="reg</pre>	'using regular expressions. ular expression"') # Search f	for substrings that match the	regular expression.
<pre># Numeric value compari e_search('count: [100, e_search('count: [*, 20 e_search('count: [200, e_search('age >= 18') e_search('age > 18')</pre>	<pre>son 200]') # >=100 and <=200 0]') # <=200 *]') # >=200 # >= 18 # >= 18</pre>		
<pre># Relational operators e_search("abc OR xyz") e_search("abc and (xyz e_search("abc ad not (e_search("abc ac axyz") e_search("abc xyz") e_search("abc !xyz")</pre>	<pre># The relational operator is or zzz)") xyz and not zzz)") # and # or # or not</pre>	s case-insensitive.	
e_match			
Ine e_match function checkSyntax	<s a="" fiel<="" log="" of="" th="" the="" value="" wnether=""><td>a meets the conditions specifie</td><td>ea in an expression.</td></s>	a meets the conditions specifie	ea in an expression.
e_match(key, regular_ex	pression, full=True)		
⑦ Note In most cas	ses, the e_match function is used	d together with the <code>op_not</code> ,	op_and , or op_or function.

• Parameters

Parameter	Туре	Required	Description
key	String	Yes	The name of the log field. If the specified field does not exist, the field does not meet the specified condition. For example, if the fl field does not exist, the function returns False.
regular_expression	String	Yes	The regular expression. If you want to match strings by using string literals, you can use the <pre>str_regex_escape</pre> function to escape characters.

full	Bool	No	Specifies whether to perform an exact match. The default value True specifies an exact match. For more information, see Regular expressions.
------	------	----	--

Response

If the specified field matches the regular expression, True is returned. Otherwise, False is returned.

Example

Check whether the value of the k1 field is a number.

Raw log

k1: 123

• Transformation rule

e_set("match",e_match("k1",r'\d+'))

Result

k1: 123 match: True

e_match_any

The e_match_any function checks whether the value of a log field meets the conditions specified in a regular expression. If one or more specified fields match the regular expression, True is returned. Otherwise, False is returned.

• Syntax

e_match_any(key1, regular_expression1, key2, regular_expression2, ..., full=True)

? Note

• The key and regular_expression parameters must be specified in pairs.

• In most cases, the e_match_any function is used together with the <code>op_not</code> , <code>op_and</code> , or <code>op_or</code> function.

• Parameters

Parameter	Туре	Required	Description
key	String	Yes	The name of the log field. If the specified field does not exist, the field does not meet the specified condition. For example, if the f1 field does not exist, the e_match_any("f1",) function returns False.
regular_expression	String	Yes	The regular expression. If you want to match strings by using string literals, you can use the str_regex_escape function to escape characters.
full	Bool	No	Specifies whether to perform an exact match. The default value True specifies an exact match. For more information, see Regular expressions.

Response

If the specified field matches the regular expression, True is returned. Otherwise, False is returned.

Example

Check whether the value of a log field meets the conditions specified in a regular expression. If one or more specified fields match the regular expression, True is returned.

Raw log

k1: 123 k2: abc k3: abc123

Transformation rule

e_set("match",e_match_any('k1', r'\d+', 'k2', '.+'))

```
• Result
```

k1:123 k2:abc k3:abc123 match:true

e_match_all

The e_match_all function checks whether the value of a log field meets the conditions specified in a regular expression. If all specified fields match the regular expression, True is returned. Otherwise, False is returned.

Syntax

e_match_all(key1, regular_expression1, key2, regular_expression2, ..., full=True)

? Note

• The key and regular_expression parameters must be specified in pairs.

 \circ In most cases, the e_match_all function is used together with the <code>op_not</code> , <code>op_and</code> , <code>or</code> <code>op_or</code> function.

• Parameters

Parameter	Туре	Required	Description
key	String	Yes	The name of the log field. If the specified field does not exist, the field does not meet the specified condition. For example, if the f1 field does not exist, the e_match_all("f1",) function returns False.
regular_expression	String	Yes	The regular expression. If you want to match strings by using string literals, you can use the <pre>str_regex_escape</pre> function to escape characters.
full	Bool	No	Specifies whether to perform an exact match. The default value True specifies an exact match. For more information, see Regular expressions.

- Response
- If the specified field matches the regular expression, True is returned. Otherwise, False is returned.
- Example
- Raw log

k1: 123 k2: abc k3: abc123

• Transformation rule

e_set("match", e_match_all("k1", r"\d+", "k2", r"\d+"))

• Result

k1:123 k2:abc k3:abc123 match:false

4.5.8.7.3. Operator functions

This topic describes the syntax and parameters of operator functions. This topic also provides examples on how to use the functions.

Functions

Category	Function	Description
	op_if	Returns the value of an expression based on a specified condition.
	op_ifnull and op_coalesce	Return the value of the first expression whose value is not None.
	op_nullif	Returns none if the value of Expression 1 is equal to the value of Expression 2. If the values of Expression 1 and Expression 2 are different, the value of Expression 1 is returned.
Conditional functions and logical functions	op_and	Evaluates the specified expressions by using the logical AND operator and returns True if all specified expressions evaluate to true. The value of each expression can be of an arbitrary data type.
	op_not	Evaluates a specified expression by using the logical NOT operator and returns the inverse Boolean value of the specified expression. The value of the expression can be of an arbitrary data type.
	op_or	Evaluates the specified expressions by using the logical OR operator, and returns True if a specified expression evaluates to true or returns False if all specified expressions evaluate to false. The value of each expression can be of an arbitrary data type.
	op_eq	Returns True or False based on the $a==b$ condition. The data types of a and b must be the same. For example, a and b are both strings, numbers, or lists.
	ob ⁻ ðe	Returns True or False based on the $a \ge b$ condition. The data types of a and b must be the same. For example, a and b are both strings, numbers, or lists.
	op_gt	Returns True or False based on the a>b condition. The data types of a and b must be the same. For example, a and b are both strings, numbers, or lists.
Comparison functions	op_le	Returns True or False based on the $a \le b$ condition. The data types of a and b must be the same. For example, a and b are both strings, numbers, or lists.

	op_lt	Returns True or False based on the $a < b$ condition. The data types of a and b must be the same. For example, a and b are both strings, numbers, or lists.
	op_ne	Returns True or False based on the $a!=b$ condition. The data types of a and b must be the same. For example, a and b are both strings, numbers, or lists.
	op_len	Calculates the number of characters in a text string. This function applies to strings or expressions that return tuples, lists, or dictionaries.
	op_in	Checks whether a string, tuple, list, or dictionary contains a specified element and returns True or False.
Container functions	op_not_in	Checks whether a string, tuple, list, or dictionary does not contain a specified element and returns True or False.
	op_slice	Extracts strings from a specified string, array, or tuple.
	op_index	Returns the element that corresponds to the index of a specified string, array, or tuple.
	op_add	Calculates the sum of multiple values. The values can be strings or numbers.
General-purpose multivalued functions	op_max	Returns the largest value among the values of multiple fields or expressions.
	op_min	Returns the smallest value among the values of multiple fields or expressions.

op_if

The op_if function returns the value of an expression based on a specified condition.

Syntax

op_if(condition, expression1, expression2)

• Parameters

Parameter	Туре	Required	Description
condition	Arbitrary	Yes	The condition. If the value of the condition is not a Boolean value, the system evaluates whether the condition is true or false. For more information, see Basic syntax.
expression1	Arbitrary	Yes	The expression whose value is returned if the evaluation result is True.
expression2	Arbitrary	Yes	The expression whose value is returned if the evaluation result is False.

Response

The value of an expression is returned.

• Examples

- Example 1: If the value of the content field evaluates to True, assign the value of Expression 1 to the test_if field.
 - Raw log

content: hello

Transformation rule

e_set("test_if", op_if(v("content"),"still origion content","replace this"))

Result

content: hello

test_if: still origion content

• Example 2: If the value of the content field evaluates to False, assign the value of Expression 2 to the test_if field.

Raw log

content: 0

Transformation rule

e_set("test_if", op_if(ct_int(v("content", default=0)),"still origion content","replace this"))

Result

content: 0
test_if: replace this

op_ifnull

The op_ifnull function returns the value of the first expression whose value is not None.

• Syntax

op_ifnull(expression1, expression2,)

• Parameters

Parameter	Туре	Required	Description
expression1	Arbitrary	Yes	Expression 1
expression2	Arbitrary	Yes	Expression 2

Response

The value of the first expression whose value is not None is returned.

- Examples
 - Example 1:
 - Raw log

test_if: hello escape_name: Etl

Transformation rule

e_set("test_ifnull", op_ifnull(v("escape_name"),v("test_if")))

Result

test_if: hello
escape_name: Etl
test_ifnull: Etl

• Example 2:

Raw log
 test_if: hello

escape_name: Etl

Transformation rule

e_set("test_ifnull", op_ifnull(v("test_if"),v("escape_name")))

Result

test_if: hello
escape_name: Etl
test_ifnull: hello

op_coalesce

The op_coalesce function returns the value of the first expression whose value is not None.

The parameters and examples of the op_coalesce function are similar to the parameters and examples of the op_ifnul1 function.

op_nullif

The op_nullif function returns none if the value of Expression 1 is equal to the value of Expression 2. If the values of Expression 1 and Expression 2 are different, the value of Expression 1 is returned.

Syntax

op_nullif(expression1, expression2)

• Parameters

Parameter	Туре	Required	Description
expression1	Arbitrary	Yes	Expression 1
expression2	Arbitrary	Yes	Expression 2

Response

None is returned if the value of Expression 1 is equal to the value of Expression 2. The value of Expression 1 is returned if the value of Expression 1 is not equal to the value of Expression 2.

- Examples
 - Example 1:
 - Raw log

content: hello escape_name: Etl

Transformation rule

e_set("test_ifnull", op_nullif(v("content"),v("escape_name")))

Result

content: hello
escape_name: Etl
test_nullif: hello

- Example 2:
 - Raw log

content: hello escape_name: hello

Transformation rule

e_set("test_ifnull", op_nullif(v("content"),v("escape_name")))

Result

In this example, the value of the content field is the same as the value of the escape_name field. Therefore, no value is assigned to the test_isnull field. content: hello

escape_name: hello

op_and

The op_and function evaluates the specified expressions by using the logical AND operator and returns True if all specified expressions evaluate to true. The value of each expression can be of an arbitrary data type.

• Syntax

op_and(value1, value2, ...)

• Parameters

Parameter	Туре	Required	Description
value1	Arbitrary	Yes	Expression 1
value2	Arbitrary	Yes	Expression 2

- Response
 - True is returned if all specified expressions evaluate to true.

• The value of each expression can be of an arbitrary type. For more information, see True or false evaluation.

- Examples
 - Example 1:
 - Raw log

number1: 123 number2: 234

Transformation rule

e_set("op_and", op_and(v("number1"),v("number2")))

Result

number1: 123 number2: 234 op_and: True

• Example 2:

Raw log

number1: 0 number2: 234

Transformation rule

e_set("op_and", op_and(v("number1"),v("number2")))

Result

number1: 0 number2: 234 op_and: True

• Example 3:

Raw log

ctx1: False ctx2: 234

Transformation rule

e_set("op_and", op_and(v("ctx1"),v("ctx2")))

Result

ctx1: False ctx2: 234 op_and: False

- Example 4:
 - Raw log

ctx1: True ctx2: 234

Transformation rule

e_set("op_and", op_and(v("ctx1"),v("ctx2")))

```
    Result
    ctx1: True
```

ctx2: 234 op_and: True

op_not

The op_not function evaluates a specified expression by using the logical NOT operator and returns the inverse Boolean value of the specified expression. The value of the expression can be of an arbitrary data type.

Syntax

op_not(expression)

• Parameters

Parameter	Туре	Required	Description
expression	Arbitrary	Yes	The expression.

Response

- The value of the expression can be of an arbitrary type. For more information, see True or false evaluation.
- Examples
 - Example 1:
 - Raw log

ctx1: True

Transformation rule

e_set("op_not", op_not(v("ctx1")))

Result

ctx1: True op_not: False

- Example 2:
 - Raw log

ctx1: 345

Transformation rule

e_set("op_not", op_not(v("ctx1")))

Result

ctx1: 345 op_not: False

• Example 3:

Raw log

ctx1: 0

Transformation rule

e_set("op_not", op_not(ct_int(v("ctx1"))))

Result

ctx1: 0 op_not: True

• Example 4:

Raw log

ctx1: ETL

Transformation rule

e_set("op_not", op_not(v("ctx1")))

Result

ctx1: ETL op_not: False

 $[\]circ\;$ The inverse Boolean value of the specified expression is returned.

- Example 5:
 - Raw log
 ctx1: None
 - Transformation rule

e_set("op_not", op_not(v("ctx1")))

Result

ctx1: None op_not: True

op_or

The op_or function evaluates the specified expressions by using the logical OR operator, and returns True if a specified expression evaluates to true or returns False if all specified expressions evaluate to false. The value of each expression can be of an arbitrary data type.

Syntax

op_or(expression1, expression2, ...)

Parameters

Parameter	Туре	Required	Description
expression1	Arbitrary	Yes	Expression 1
expression2	Arbitrary	Yes	Expression 2

Response

• True is returned if a specified expression evaluates to true. False is returned if all specified expressions evaluate to false.

- The value of each expression can be of an arbitrary type. For more information, see True or false evaluation.
- Examples
 - Example 1:
 - Raw log

ctx1: 123 ctx2: 234

Transformation rule

e_set("op_or", op_or(v("ctx1"),v("ctx2")))

Result

ctx1: 123 ctx2: 234 op_or: True

- Example 2:
 - Raw log ctx1: 0
 - ctx2: 234
 - Transformation rule

e_set("op_or", op_or(v("ctx1"),v("ctx2")))

Result

ctx1: 0 ctx2: 234 op_or: True

• Example 3:

Raw log
 ctx1: ETL

ctx2: ALIYUN

Transformation rule

e_set("op_or", op_or(v("ctx1"),v("ctx2")))

Result

ctx1: ETL ctx2: ALIYUN op_or: True

- Example 4:
 - Raw log

ctx1: True ctx2: False

Transformation rule

e_set("op_or", op_or(v("ctx1"),v("ctx2")))

Result

ctx1: True ctx2: False op_or: True

• Example 5:

Raw log

ctx1: 0 ctx2: False

Transformation rule

e_set("op_or", op_or(ct_int(v("ctx1")),v("ctx2")))

Result

```
ctx1: 0
ctx2: False
op_or: False
```

• Example 6:

Raw log

ctx1: 124 ctx2: True

Transformation rule

e_set("op_or", op_or(v("ctx1"),v("ctx2")))

Result

ctx1: 124 ctx2: True op_or: True

op_eq

The op_eq function returns True or False based on the a==b condition.

Syntax

op_eq(value1, value2)

• Parameters

Parameter	Туре	Required	Description
value1	Arbitrary	Yes	Expression 1
value2	Must be the same as the data type of Expression 1	Yes	Expression 2

Response

True is returned if the value of Expression 1 is equal to the value of Expression 2. False is returned if the value of Expression 1 is not equal to the value of Expression 2.

- Examples
 - Example 1:
 Raw log

```
content: hello
ctx: hello
```

Transformation rule

e_set("test_eq", op_eq(v("content"),v("ctx")))

```
    Result
```

content: hello ctx: hello test_eq: True

- Example 2:
 - Raw log

content: hello ctx: ctx

Transformation rule

e_set("test_eq", op_eq(v("content"),v("ctx")))

```
    Result
```

content: hello ctx: ctx test eq: False

op_ge

The op_ge function returns True or False based on the $a \ge b$ condition.

• Syntax

op_ge(value1, value2)

• Parameters

Parameter	Туре	Required	Description
value1	Arbitrary	Yes	Expression 1
value2	Must be the same as the data type of Expression 1	Yes	Expression 2

Response

True is returned if the value of Expression 1 is greater than or equal to the value of Expression 2. False is returned if the value of Expression 1 is less than the value of Expression 2.

- Examples
 - Example 1: Return True if the value of the apple_price field is greater than or equal to the value of the orange_price field.

Raw log apple_price: 16

orange_price: 14

Transformation rule

e_set("test_ge", op_ge(ct_int(v("apple_price")),ct_int(v("orange_price"))))

```
    Result
```

apple_price: 16 orange_price: 14 test_ge: True

 \circ Example 2: Return False if the value of the apple_price field is less than the value of the orange_price field.

Raw log

apple_price: 12 orange_price: 14

Transformation rule

e_set("test_ge", op_ge(ct_int(v("apple_price")),ct_int(v("orange_price"))))

Result

apple_price: 12 orange_price: 14 test ge: False

op_gt

The op_gt function returns True or False based on the \$a>b\$ condition.

Syntax

op_gt(value1, value2)

• Parameters

Parameter	Туре	Required	Description
value1	Arbitrary	Yes	Expression 1
value2	Must be the same as the data type of Expression 1	Yes	Expression 2

Response

True is returned if the value of Expression 1 is greater than the value of Expression 2. False is returned if the value of Expression 1 is less than or equal to the value of Expression 2.

Examples

- Example 1: Return True if the value of the old_number field is greater than the value of the young_number field. Return False if the value of the old_number field is less than or equal to the value of the young_number field.
 - Raw log

old_number: 16 young_number: 14

Transformation rule

e_set("op_gt",op_gt(ct_int(v("old_number")),ct_int(v("young_number"))))

Result

old_number: 16 young_number: 14 test_ge: True

• Example 2: Return True if the value of the priority field is greater than the value of the price field. Return False if the value of the priority field is less than or equal to the value of the price field.

Raw log

- priority: 14 price: 16
- Transformation rule

e_set("op_gt",op_gt(ct_int(v("priority")),ct_int(v("price"))))

Result

priority: 14 price: 16 test_ge: False

op_le

The op_le function returns True or False based on the \$a<=b\$ condition.

Syntax

op_le(value1, value2)

Parameters

Parameter	Туре	Required	Description
value1	Arbitrary	Yes	Expression 1
value2	Must be the same as the data type of Expression 1	Yes	Expression 2

Response

True is returned if the value of Expression 1 is less than or equal to the value of Expression 2. False is returned if the value of Expression 1 is greater than the value of Expression 2.

- Examples
- Example 1: Return True if the value of the priority field is less than or equal to the value of the price field. Return False if the value of the priority field is greater than the value of the price field.
 - Raw log
 priority: 16
 - price: 14
 - Transformation rule

e_set("op_le", op_le(ct_int(v("priority")), ct_int(v("price"))))

Result

priority: 16
price: 14
test_ge: False

• Example 2: Return True if the value of the priority field is less than or equal to the value of the price field. Return False if the value of the priority field is greater than the value of the price field.

```
    Raw log
```

priority: 14 price: 16

Transformation rule

e_set("op_le", op_le(ct_int(v("priority")), ct_int(v("price"))))

Result

priority: 14 price: 16 test_ge: True

op_lt

The op_lt function returns True or False based on the a
b condition.

• Syntax

op_lt(value1, value2)

Parameters

Parameter	Туре	Required	Description
value1	Arbitrary	Yes	Expression 1
value2	Must be the same as the data type of Expression 1	Yes	Expression 2

Response

True is returned if the value of Expression 1 is less than the value of Expression 2. False is returned if the value of Expression 1 is greater than or equal to the value of Expression 2.

Examples

• Example 1: Return True if the value of the priority field is less than the value of the price field. Return False if the value of the priority field is greater than or equal to the value of the price field.

- Raw log
 priority: 16
- price: 14
- Transformation rule

e_set("op_lt",op_lt(ct_int(v("priority")),ct_int(v("price"))))

Result

priority: 16
price: 14
op_lt: False

- Example 2: Return True if the value of the priority field is less than the value of the price field. Return False if the value of the priority field is greater than or equal to the value of the price field.
 - Raw log

priority: 14 price: 15

Transformation rule

e_set("op_lt",op_lt(ct_int(v("priority")),ct_int(v("price"))))

Result

priority: 14 price: 15 op_lt: True

op_ne

The op_ne function returns True or False based on the a!=b condition.

• Syntax

op_ne(value1, value2)

• Parameters

Parameter	Туре	Required	Description
value1	Arbitrary	Yes	Expression 1
value2	Must be the same as the data type of Expression 1	Yes	Expression 2

Response

True is returned if the value of Expression 1 is not equal to the value of Expression 2. False is returned if the value of Expression 1 is equal to the value of Expression 2.

- Examples
 - Example 1:
 - Raw log

priority: 16 price: 14

Transformation rule

e_set("op_ne", op_ne(ct_int(v("priority")), ct_int(v("price"))))

Result

priority: 16 price: 14 op_ne: True

- Example 2:
 - Raw log

priority: 14 price: 14

Transformation rule

e_set("op_ne",op_ne(ct_int(v("priority")),ct_int(v("price"))))

```
    Result
```

priority: 14
price: 14
op_ne: False

op_len

The op_len function calculates the number of characters in a text string. This function applies to strings or expressions that return tuples, lists, or dictionaries.

Syntax

op_len(value)

Parameters

Parameter	Туре	Required	Description
value	String, tuple, list, or dictionary	Yes	The expression.

- Response
- The length of the specified expression value is returned.
- Example
- Raw log
 - content: I,love,this,world
- Transformation rule

e_set("op_len", op_len(v("content")))

• Result

content: I,love,this,world
op_len: 17

op_in

The op_in function checks whether a string, tuple, list, or dictionary contains a specified element and returns True or False.

Syntax

op_in(value1, value2)

• Parameters

Parameter	Туре	Required	Description
value1	String, tuple, list, or dictionary	Yes	The string, tuple, list, or dictionary.
value2	Arbitrary	Yes	The element that you want to check.

(?) Note In a function, the specified string, tuple, list, or dictionary is placed before the specified element.

Response

True is returned if the specified string, tuple, list, or dictionary contains the specified element. False is returned if the specified string, tuple, list, or dictionary does not contain the specified element.

Example
 Raw log

```
list: [1, 3, 2, 7, 4, 6]
```

Transformation rule

num2: 2

e_set("op_in", op_in(v("list"), v("num2")))

• Result

list: [1, 3, 2, 7, 4, 6] num2: 2 op_in: True

op_not_in

The op_not_in function checks whether a string, tuple, list, or dictionary does not contain a specified element and returns True or False.

Syntax

op_not_in(value1, value2)

• Parameters

Parameter	Туре	Required	Description
value1	String, tuple, list, or dictionary	Yes	The string, tuple, list, or dictionary.
value2	Arbitrary	Yes	The element that you want to check.

③ Note In a function, the specified string, tuple, list, or dictionary is placed before the specified element.

Response

True is returned if the specified string, tuple, list, or dictionary does not contain the specified element. False is returned if the specified string, tuple, list, or dictionary contains the specified element.

Example

```
• Raw log
list: [1, 3, 2, 7, 4, 6]
num2: 12
```

Transformation rule

e_set("op_not_in", op_not_in(v("list"), v("num2")))

• Result

```
list: [1, 3, 2, 7, 4, 6]
num2: 12
op_in: True
```

op_slice

The op_slice function extracts strings from a specified string, array, or tuple.

- Syntax
 - op_slice(value, start, end=None, step=None)

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value that you want to truncate.
start	Num	No	The position from which the value is truncated. By default, the truncation starts from the first character of the string.
end	Num	No	The position to which the value is truncated. The character at this position is not included. The default value is the end of the specified string.
step	Num	No	The step that is used for truncating.

Response

The string that is extracted from the truncated value is returned.

- Examples
 - $\circ~$ Example 1: Truncate the value of the word field from the beginning to the end at a step of 2.
 - Raw log

word: I,love,this,world

Transformation rule

e_set("op_slice", op_slice(v("word"), 2))

Result

word: I,love,this,world
op_slice: I,

- $\circ~$ Example 2: Truncate the value of the word field from position 2 to position 9 at a step of 1.
- Raw log
 - word: I,love,this,world

```
    Transformation rule
```

e_set("op_slice", op_slice(v("word"), 2, 9, 1))

Result

word: I,love,this,world
op_slice: love,th

op_index

The op_index function returns the element that corresponds to the index of a specified string, array, or tuple.

Syntax

op_index(value, index)

Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value that you want to truncate.
index	Num	No	The index of the specified string, array, or tuple.

- Response
- The element that corresponds to the index is returned.
- Examples
 - Example 1: Obtain the element that corresponds to the index 0 in the value of the word field.
 - Raw log
 - word: I,love,this,world
 - Transformation rule

e_set("op_index", op_index(v("word"), 0))

- Result
 - word: I,love,this,world
 op_slice: I
- Example 2: Obtain the element that corresponds to the index 3 in the value of the word field.

Raw log

word: I,love,this,world

Transformation rule

e_set("op_index", op_index(v("word"),3))

Result

word: I,love,this,world
op_index: o

op_add

The op_add function calculates the sum of multiple values. The values can be strings or numbers.

Syntax

op_add(value1, value2, ...)

• Parameters

Parameter	Туре	Required	Description
value1	String, tuple, list, or dictionary	Yes	Value 1
value2	Must be the same as the data type of Value 1	Yes	Value 2

Response

The sum of the specified values is returned.

- Examples
 - Example 1: Calculate the sum of the values of the price_orange and price_apple fields.
 - Raw log

price_orange: 2
price_apple: 13

Transformation rule

e_set("account", op_add(ct_int(v("price_orange")), ct_int(v("price_apple"))))

Result

price_orange: 2
price_apple: 13
account: 15

- Example 2: Calculate the sum of the values of the bytes_in and bytes_out fields.
- Raw log

bytes_in: 214 bytes_out: 123

Transformation rule

e_set("total_bytes", op_add(ct_int(v("bytes_in")), ct_int(v("bytes_out"))))

Result

bytes_in: 214 bytes_out: 123 total bytes: 337

• Example 3: Add the https:// prefix to a URL.

Raw log

host: aliyun.com

Transformation rule

e_set("website", op_add("https://", v("host")))

Result

host: aliyun.com website: https://aliyun.com

op_max

The op_max function returns the largest value among the values of multiple fields or expressions.

• Syntax

op_max(value1, value2, ...)

• Parameters

Parameter	Туре	Required	Description
valuel	Arbitrary	Yes	Value 1
value2	Must be the same as the data type of Value 1	Yes	Value 2

Response

The largest value among the specified values is returned.

• Example

• Raw log

priority_apple: 13

• Transformation rule

e_set("max_price", op_max(ct_int(v("price_orange")),ct_int(v("priority_apple"))))

• Result

```
price_orange: 2
priority_apple: 13
max_price: 13
```

op_min

The op_min function returns the smallest value among the values of multiple fields or expressions.

Syntax

op_min(value1, value2, ...)

• Parameters

Parameter	Туре	Required	Description
value1	Arbitrary	Yes	Value 1
value2	Must be the same as the data type of Value 1	Yes	Value 2

Response

The smallest value among the specified values is returned.

Example

```
• Raw log
```

```
price_orange: 2
price_apple: 13
```

• Transformation rule

e_set("op_min", op_min(ct_int(v("price_orange")),ct_int(v("price_apple"))))

```
• Result
```

```
price_orange: 2
price_apple: 13
op_min: 2
```

4.5.8.7.4. Conversion functions

This topic describes the syntax and parameters of conversion functions. This topic also provides examples on how to use the functions.

Functions

Category	Function	Description	
	ct_int	Converts the value of a field or an expression to an integer.	
Pacis type conversion	ct_float	Converts the value of a field or an expression to a floating-point number.	
basic type conversion	ct_str	Converts the value of a field or an expression to a string.	
	ct_bool	Converts the value of a field or an expression to a Boolean value.	
	ct_chr	Converts the ANSI or Unicode value of a field or an expression to a character.	
	ct_ord	Converts the value of a field or an expression to an ANSI value or a Unicode value.	
Number conversion	ct_hex	Converts the value of a field or an expression to a hexadecimal number.	
	ct_oct	Converts the value of a field or an expression to an octal number.	
	ct_bin	Converts the value of a field or an expression to a binary number.	
Numeral system conversion	bin2oct	Converts a binary number to an octal number.	
Numeral System Conversion	bin2hex	Converts a binary number to a hexadecimal string.	

ct_int

The ct_int function converts the value of a field or an expression to an integer.

- Syntax
- ct_int(value, base=10)

Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to convert.
base	Number	No	The numeral system. Default value: 10. This value specifies the decimal numeral system. If you set the base parameter to 8, this function converts an octal value to a decimal value.

Response

An integer is returned.

- Examples
- Example 1: Convert a string to an integer.
 - Raw log

```
number: 2
```

Transformation rule

e_set("int_number", ct_int(v("number")))

Result

```
number: 2
int_number: 2
```

• Example 2: Convert a hexadecimal value to a decimal value.

- Raw log
- number: AB
- Transformation rule

e_set("int_number", ct_int(v("number"),base=16))

Result

number: AB int_number: 171

ct_float

The ct_float function converts the value of a field or an expression to a floating-point number.

• Syntax

ct_float(value)

• Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to convert.
_			

Response

A floating-point number is returned.

- Example
- Raw log

price: 2

• Transformation rule

e_set("price_float", ct_float(v("price")))

• Result

price: 2
price_float: 2.0

ct_str

The ct_str function converts the value of a field or an expression to a string.

• Syntax

ct_str(value)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary	Yes	The value that you want to convert.

Response

A string is returned.

- Example
- Transformation rule

e_set("ct_str", ct_str(b'test byte'))

Result

ct_str: test byte

ct_bool

The ct_bool function converts the value of a field or an expression to a Boolean value. For information about the true or false evaluation of different data types, see True or false evaluation.

• Syntax

ct_bool(value)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary	Yes	The value that you want to convert.

Response

A Boolean value is returned.

- Example
- Raw log

num: 2

• Transformation rule

e_set("ct_bool", ct_bool(v("num")))

Result

num: 2 ct_bool: true

ct_chr

The ct_chr function converts the ANSI or Unicode value of a field or an expression to a character.

- Syntax
- ct_chr(value)
- Parameters

User Guide-Log Service

	Parameter	Туре	Required	Description			
	value	Number or numeric string	Yes	The value that you want to convert.			
•	Response						
	A character is returned.						
•	Example						
• Raw log number: 78							
							 Transformation rule
	<pre>e_set("ct_chr", ct_chr(v("number")))</pre>	"))))					
4	• Result						
	number: 78 ct_chr: N						
ct	t ord						
Th	 e ct ord function converts the value of 	of a field or an expression to an ANSI	value or a Unicode value.				
•	 Syntax						
	ct_ord(value)						
•	Parameters						
	Parameter	Туре	Required	Description			
	value	String	Yes	The value that you want to convert. The value contains only one character.			
•	Response						
	An ANSI value or a Unicode value is re	eturned.					
•	Example						
	• Raw log						
	world: a						
	 Transformation rule 						
	<pre>e_set("ct_ord", ct_ord(v("world")))</pre>						
	• Result	Result					
	world: a ct_ord: 97						
ct	t hex						
ть							
- 111	e ct hex function converts the value (of a field or an expression to a hexade	ecimal number.				

ct_hex(value)

• Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to convert.

Response

A hexadecimal number is returned.

- Example
- Raw log
 - number: 123

• Transformation rule

e_set("ct_hex", ct_hex(v("number")))

• Result

number: 123 ct_hex: 0x7b

ct_oct

The ct_oct function converts the value of a field or an expression to an octal number.

Syntax

ct_oct(value)

• Parameters

User Guide-Log Service

Cloud Defined Storage

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to convert.
Response			
An octal number is returned.			
Example			
• Raw log			
number: 123			

• Transformation rule

e_set("ct_oct", ct_oct(v("number")))

• Result

number: 123 ct_oct: 0o173

ct_bin

The ct_bin function converts the value of a field or an expression to a binary number.

- Syntax
 - ct_bin(value)
- Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to convert.
Response			
A binary number is returned.			

• Example

•

- Raw log
- number: 123
- Transformation rule

e_set("ct_bin", ct_bin(v("number")))

• Result

number: 123 ct_bin: 0b1111011

bin2oct

The bin2oct function converts a binary number to an octal number.

- Syntax
 - bin2oct(binary)
- Parameters

Parameter	Туре	Required	Description
binary	Binary	Yes	The binary string that you want to convert.

- Response
- An octal string is returned.
- Example
- Raw log
- test : test

e_set("new",bin2oct(base64_decoding("ARi8WnFiLAAACHcAGgkADV37Xs8BXftezgAdgwF9")))

Result

```
test : test
```

new : 21427426470542130000002073400064044000325677327547401273755366340003552600575

bin2hex

The bin2hex function converts a binary number to a hexadecimal string.

• Syntax

bin2hex(binary)

• Parameters

[•] Transformation rule

User Guide-Log Service

Parameter	Туре	Required	Description
binary	Binary	Yes	The binary string that you want to convert.

- Response
- A hexadecimal string is returned.
- Example
- Raw log
- test : test
- Transformation rule

e_set("new",bin2hex(base64_decoding("ARi8WnFiLAAACHcAGgkADV37Xs8BXftezgAdqwF9"))))

- Result
- test : test

new :0118bc5a71622c00000877001a09000d5dfb5ecf015dfb5ece001dab017d

4.5.8.7.5. Arithmetic functions

This topic describes the syntax and parameters of arithmetic functions. This topic also provides examples on how to use the functions.

unctions				
⑦ Note If you want to pase	ss a negative integer, use <code>op_neg(P</code>	Positive integer) . For example, use $op_neg(1)$ to indicate -1 .		
Category	Function	Description		
Sum calculation	op_sum	Returns the sum of input values.		
	op_abs	Returns the absolute value of an input value.		
	op_div_floor	Returns the integer part of the quotient of two input values.		
	op_div_true	Returns the quotient of two input values.		
Basic calculation	op_pow	Returns a value raised to a specified power.		
	op_mul	Returns the product of two input values.		
	op_neg	Returns the opposite number of an input value.		
	op_mod	Returns the remainder of an input value divided by the other input value.		
	op_sub	Returns the difference between two input values.		
	op_round	Rounds an input value.		
	mat_ceil	Rounds an input value rounded up to the nearest integer.		
	mat_exp	Returns Euler's number raised to the power of an input value.		
	mat_fabs	Returns the absolute value of an input value.		
	mat_floor	Rounds an input value down to the nearest integer.		
	mat_log	Returns the logarithm of an input value with the base specified by the other input value.		
	mat_log10	Returns the base-10 logarithm of an input value.		
	mat_sqrt	Returns the square root of an input value.		
	mat_degrees	Converts radians to degrees.		
	mat_radians	Converts degrees to radians.		
	mat_sin	Returns the sine of an input value in radians.		
lathematical calculation	mat_cos	Returns the cosine of an input value in radians.		
	mat_tan	Returns the tangent of an input value in radians.		
	mat_acos	Returns the arc cosine of an input value in radians.		
	mat_asin	Returns the arc sine of an input value in radians.		
	mat_atan	Returns the arc tangent of an input value in radians.		
	mat_atan2	Returns the arc tangent of X and Y coordinates.		
	mat_atanh	Returns the inverse hyperbolic tangent of an input value.		
	mat_hypot	Returns the Euclidean norm of two input values.		
	MATH_PI	Obtains the constant pi.		

obtains the constance.	M	MATCH_E	Obtains the constant e.
------------------------	---	---------	-------------------------

op_sum

The op_sum function returns the sum of input values.

```
    Syntax
```

op_sum(value1, value2, ...)

• Parameters

Parameter	Туре	Required	Description
valuel	Number or numeric string	Yes	The value that you want to use for the calculation.
value2	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The sum of all input values is returned.

• Example

Calculate the sum of the values of the course_price and goods_price fields.

Raw log

course_price: 12 goods_price: 2

• Transformation rule

e_set("account", op_sum(v("course_price"), v("goods_price")))

• Result

```
course_price: 12
goods_price: 2
account: 14
```

op_abs

The op_abs function returns the absolute value of an input value.

Syntax

op_abs(value)

Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The absolute value of the input value is returned.

• Example

Calculate the absolute value of the value of the course_price field.

- Raw log
 - course_price: -4
- Transformation rule

e_set("op_abs", op_abs(v("course_price")))

Result

course_price: -4 op_abs: 4

op_div_floor

The op_div_floor function returns the integer part of the quotient of two input values.

• Syntax

op_div_floor(value1, value2)

• Parameters

Parameter	Туре	Required	Description
value1	Number or numeric string	Yes	The value that you want to use for the calculation.
value2	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The integer part of the quotient that is obtained after value1 is divided by value2 is returned.

• Example

Calculate the unit price based on the values of the course_price and count fields.

```
    Raw log
```

course_price: 4 count: 2

Transformation rule

e_set("op_div_floor", op_div_floor(v("course_price"), v("count")))

```
• Result
```

course_price: 4 count: 2 op_div_floor: 2

op_div_true

The op_div_true function returns the quotient of two input values.

? Note This function automatically converts the data types of input values. The input value can be a string or an integer.

• Syntax

op_div_true(value1, value2)

• Parameters

Parameter	Туре	Required	Description
value1	Number or numeric string	Yes	The value that you want to use for the calculation.
value2	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The quotient that is obtained after value1 is divided by value2 is returned.

```
    Examples
```

 $\circ~$ Example 1: Calculate the unit price based on the values of the fruit_price and count fields.

Raw	log			

```
fruit_price: 9
count: 2
```

Transformation rule

e_set("op_div_true", op_div_true(v("fruit_price"), v("count")))

Result

fruit_price: 9
count: 2
op_div_true: 4.5

- Example 2: Calculate the acceleration based on the values of the one_speed and two_speed fields. The return value is rounded. Formula: a = (one_speed two_speed) / time .
 - Raw log

one_speed: 9 two_speed: 2 time: 3

Transformation rule

e_set("a", op_round(op_div_true(op_sub(v("one_speed"), v("two_speed")), v("time")), 2))

Result

a:2.33 one_speed:9 time:3 two_speed:2

op_pow

The op_pow function returns a value raised to a specified power.

• Syntax

op_pow(value1, value2)

• Parameters

Parameter	Туре	Required	Description
valuel	Number or numeric string	Yes	The value that you want to use for the calculation.

User Guide-Log Service

Cloud Defined Storage

value2 Number or numeric string	Yes	The value that you want to use for the calculation.
---------------------------------	-----	---

Response

value1 raised to the power of value2 is returned.

• Example

Calculate the value of the course field raised to the power of the value of the pow field.

с

course: 100

pow: 2

• Transformation rule

e_set("pow_course", op_pow(v("course"), v("pow")))

• Result

course: 100 pow: 2 pow_course: 10000

op_mul

The op_mul function returns the product of two input values.

Syntax

op_mul(value1, value2)

• Parameters

Parameter	Туре	Required	Description
value1	Number, string, tuple, or list	Yes	The value that you want to use for the calculation.
value2	Number	Yes	The value that you want to use for the calculation.

Response

- If value1 and value2 are numbers, the product of value1 and value2 is returned.
- If value1 and value2 are strings, tuples, or lists, the specified number of duplicates of the original value is returned.
- Examples
 - Example 1: Calculate the product of the values of the course and price fields.
 - Raw log

course: 10 price: 23

Transformation rule

e_set("account", op_mul(ct_int(v("course")), ct_int(v("price"))))

Result

course: 10 price: 23 account: 230

• Example 2: Return three duplicates of the original value of the course field.

- Raw log
 - course: "abc"
- Transformation rule

e_set("course", op_mul(v("course"), 3))

Result

course: "abcabcabc"

op_neg

The op_neg function returns the opposite number of an input value.

Syntax

op_neg(value)

• Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The opposite number of the input value is returned.

- Example
 - Raw log
 - course: -100
 - Transformation rule

e_set("account", op_neg(v("course_price")))

• Result

course: -100 account: 100

op_mod

The op_mod function returns the remainder of an input value divided by the other input value.

Syntax

op_mod(value1, value2)

Parameters

Parameter	Туре	Required	Description
valuel	Number or numeric string	Yes	The value that you want to use for the calculation.
value2	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The remainder that is obtained after value1 is divided by value2 is returned.

- Example
 - Raw log

course: 4 count: 3

Transformation rule

e_set("op_mod", op_mod(v("course"), v("count")))

• Result

```
course: 4
count: 3
op_mod: 1
```

op_sub

The op_sub function returns the difference between two input values.

Syntax

op_sub(value1, value2)

• Parameters

Parameter	Туре	Required	Description
valuel	Number or numeric string	Yes	The value that you want to use for the calculation.
value2	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The difference between value1 and value2 is returned.

• Example

Calculate the difference between the values of the count and count_apple fields.

- Raw log
 - count: 6 count_apple: 3
- Transformation rule

e_set("sub_number", op_sub(v("count"),v("count_apple")))

Result

```
count: 6
count_apple: 3
sub_number: 3
```

op_round

The op_round function rounds an input value.

Syntax

op_round(value, number)

Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.
number	Number	Yes	The number of decimal places to which the value is rounded. Default value: 0.

- Response
- The rounded input value is returned.
- Example

Round the value of the price field to one decimal place.

- price: 4.56
- Transformation rule
 - e_set("round_price", op_round(v("price"),1))
- Result

price: 4.56

round_price: 4.6

mat_ceil

The mat_ceil function rounds an input value up to the nearest integer.

Syntax

mat_ceil(value)

• Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The smallest integer that is not less than the input value is returned.

- Example
- Raw log
 - price: 4.1
- Transformation rule

e_set("mat_ceil", mat_ceil(v("price")))

• Result

price: 4.1

mat_ceil: 5

mat_exp

The mat_exp function returns Euler's number raised to the power of an input value.

• Syntax

mat_exp(value)

• Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

Euler's number raised to the power of the input value is returned.

- Example
 - Raw log
 - number: 2
 - Transformation rule

e_set("e_x", mat_exp(v("number")))

• Result

number: 1 e_x: 7.38905609893065

mat_fabs

The mat_fabs function returns the absolute value of an input value.

- Syntax
 - mat_fabs(value)

• Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The absolute value of the input value is returned.

- Example
- Raw log
- course_price: -10
- Transformation rule

e_set("mat_fabs", mat_fabs(v("course_price")))

```
• Result
```

course_price: -10
mat_fabs: 10.0

mat_floor

The mat_floor function rounds an input value down to the nearest integer.

Syntax

```
mat_floor(value)
```

```
• Parameters
```

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The largest integer that is not greater than the input value is returned.

- Example
- Raw log

course_price: 4.9

• Transformation rule

e_set("mat_floor", mat_floor(v("course_price")))

• Result

course_price: 4.9

mat_floor: 4

mat_log

The mat_log function returns the logarithm of an input value with the base specified by the other input value.

• Syntax

mat_log(value1,value2)

• Parameters

Parameter	Туре	Required	Description
value1	Number or numeric string	Yes	The value that you want to use for the calculation.
value2	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The logarithm of value1 with base value2 is returned.

• Example

• Raw log

number1: 100 number2: 10

• Transformation rule

e_set("mat_log", mat_log(v("number1"),v("number2")))

• Result

number1: 100 number2: 10 mat_log: 2.0

mat_log10

The mat_log10 function returns the base-10 logarithm of an input value.

• Syntax

mat_log10(value)

• Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The base-10 logarithm of the input value is returned.

- Example
- Raw log

number: 100

• Transformation rule

e_set("number2", mat_log10(v("number")))

• Result

number: 100 numbe2: 2.0

mat_sqrt

The mat_sqrt function returns the square root of an input value.

- Syntax
 - mat_sqrt(value)

Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The square root of the input value is returned.

- Example
- Raw log
- number1: 100
- Transformation rule

e_set("sqrt_account", mat_sqrt(v("number1")))

• Result

number1: 100 sqrt_account: 10.0

mat_degrees

The mat_degrees function converts radians to degrees.

Syntax

mat_degrees(value)

• Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The value in degrees is returned.

- Example
 - Raw log
 - Transformation rule

e_set("mat_degrees", mat_degrees(v("num")))

• Result

num: 1 mat_degrees: 57.29577951308232

mat_radians

The mat_radians function converts degrees to radians.

• Syntax

mat_radians(value)

• Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The value in radians is returned.

- Example
- Raw log
- rad: 30
- Transformation rule

e_set("mat_radians", mat_radians(v("rad")))

• Result

rad: 30 mat_radians: 0.5235987755982988

mat_sin

The mat_sin function returns the sine of an input value in radians.

Syntax

mat_sin(value)

• Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

- Response
- The sine of the input value is returned.
- Example
- Raw log
- Transformation rule

e_set("mat_sin", mat_sin(v("sin")))

• Result

sin: 90 mat_sin: 0.8939966636005579

mat_cos

The mat_cos function returns the cosine of an input value in radians.

• Syntax

- mat_cos(value)
- Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response
The cosine of the input value is returned.

- Example
 - Raw log
 - Transformation rule

e_set("mat_cos", mat_cos(v("cos")))

• Result

cos: 30 mat_cos: 0.15425144988758405

mat_tan

The mat_tan function returns the tangent of an input value in radians.

• Syntax

mat_tan(value)

• Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The tangent of the input value is returned.

- Example
- Raw log
- tan: 30
- Transformation rule

e_set("mat_tan", mat_tan(v("tan")))

• Result

tan: 30 mat_tan: 1.6197751905438615

mat_acos

The mat_acos function returns the arc cosine of an input value in radians.

Syntax

mat_acos(value)

• Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

- Response
- The arc cosine of the input value is returned.
- Example
- Raw log
- Transformation rule

e_set("mat_acos", mat_acos(v("acos")))

• Result

```
acos: 1
mat_acos: 0.0
```

mat_asin

The mat_asin function returns the arc sine of an input value in radians.

• Syntax

- mat_asin(value)
- Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The arc sine of the input value is returned.

- Example
 - Raw log
 - Transformation rule

e_set("mat_asin", mat_asin(v("asin")))

Result

asin: 1 mat_asin: 1.5707963267948966

mat_atan

The mat_atan function returns the arc tangent of an input value in radians.

• Syntax

mat_atan(value)

• Parameters

Parameter	Туре	Required	Description
value	Number or numeric string	Yes	The value that you want to use for the calculation.

Response

The arc tangent of the input value is returned.

- Example
- Raw log
- atan: 1
- Transformation rule

e_set("mat_atan", mat_atan(v("atan")))

• Result

atan: 1 mat_atan: 0.7853981633974483

mat_atan2

The mat_atan2 function returns the arc tangent of X and Y coordinates.

Syntax

mat_atan2(x,y)

• Parameters

Parameter	Туре	Required	Description
x	Number or numeric string	Yes	The X-coordinate.
У	Number or numeric string	Yes	The Y-coordinate.

Response

The arc tangent of the X-coordinate and the Y-coordinate is returned.

- Example
 Raw log
 - atan1: 1 atan2: 2
 - Transformation rule

e_set("mat_atan2", mat_atan2(v("atan1"),v("atan2")))

```
• Result
```

atan1: 1 atan2: 2 mat_atan2: 0.4636476090008061

mat_atanh

The mat_atanh function returns the inverse hyperbolic tangent of an input value.

•	Syntax			
	<pre>mat_atanh(value)</pre>			
•	Parameters			
	Parameter	Туре	Required	Description

value	Number or numeric string	Yes	The X-coordinate.	
Response The inverse hyperbolic tangent of the input value is returned. Example • Raw log				
atanh: 0.5				
Transformation rule				
<pre>e_set("mat_atanh", mat_atanh(v("a</pre>	atanh")))			
• Result				
atanh:0.5 mat_atanh:0.5493061443340548				
mat_hypot The mat_hypot function returns the Euc Syntax	lidean norm of two input values.			
<pre>mat_hypot(value1,value2)</pre>				
Parameters				
Parameter	Туре	Required	Description	
valuel	Number or numeric string	Yes	The X-coordinate.	
value2	Number or numeric string	Yes	The Y-coordinate.	
 Response The Euclidean norm of the input value Example Raw log 	es is returned.			
hypot1: 1 hypot2: 2				
 Transformation rule 				
<pre>e_set("mat_hypot", mat_hypot(v(")</pre>	nypot1"),v("hypot2")))			
• Result				
hypot1:1 hypot2:2 mat_hypot:2.23606797749979				
MATH_PI The MATH_PI function obtains the constant pi. • Raw log				
a:2				
Transformation rule				
<pre>e_set("result", op_sum(v("a"), MATH</pre>	H_PI))			
• Result				
result:5.141592653589793				
MATCH_E The MATH_E function obtains the consta • Raw log	ant e.			
a:2				
Transformation rule				
<pre>e_set("result", op_sum(v("a"), MATCH_E))</pre>				
• Result				
a:2 result:4.718281828459045	a:2 result:4.718281828459045			
4.5.8.7.6. String functions				

This topic describes the syntax and parameters of string functions. This topic also provides examples on how to use the functions.

Functions

User Guide-Log Service

Category	Function	Description
	str_format	Formats strings.
Multi-string operation	str_join	Concatenates input strings to generate a new string by using a specified connector.
	str_zip	Concurrently splits two values or strings that are returned by expressions and combines the results into one string.
	str_encode	Encodes a string by using a specified encoding format.
	str_decode	Decodes an input value by using a specified encoding format.
Encoding and decoding	str_hex_escape_encode	Escapes special characters. The function can escape hexadecimal characters to Chinese characters.
	str_uuid	Generates a random UUID.
	str_sort	Sorts a specified object.
	str_reverse	Reverses a string.
Sorting, reversing, and replacement	str_replace	Replaces an existing string with a specified string based on a specified rule.
	str_logstash_config_normalize	Converts data in the Logstash configuration language to the JSON format.
	str_translate	Replaces specified characters in a string with mapping characters.
	str_strip	Deletes specified characters from a string.
	str_lstrip	Deletes specified characters from the start of a string.
	str_rstrip	Deletes specified characters from the end of a string.
	str_lower	Converts all uppercase letters in a string to lowercase letters.
Regular munging	str_upper	Converts all lowercase letters in a string to uppercase letters.
	str_title	Capitalizes the first letter of each word in a string and converts the other letters in the string to lowercase letters.
	str_capitalize	Capitalizes the first letter of a string and converts the other letters in the string to lowercase letters.
	str_swapcase	Converts the uppercase letters to lowercase letters and lowercase letters to uppercase letters in a string.
	str_count	Counts the number of occurrences of a character in a string.
	str_find	Checks whether a string contains a specified substring.
Search and check	str_rfind	Returns the position of the last occurrence of a specified string or a specified character in a string.
	str_endswith	Checks whether a string ends with a specified suffix.
	str_startswith	Checks whether a string starts with a specified string.
	str_split	Splits a string by using a specified delimiter.
Colitting	str_splitlines	Splits a string by using a line feed.
Spitting	str_partition	Splits a string into three parts from left to right by using a specified delimiter.
	str_rpartition	Splits a string into three parts from right to left by using a specified delimiter.
	str_center	Pads a string to a specified length by using a specified character.
	str_ljust	Pads a string to a specified length by using a specified character from the end of the string.
Formatting	str_rjust	Pads a string to a specified length by using a specified character from the start of the string.
	str_zfill	Pads a string to a specified length by using 0 from the start of the string.
	str_expandtabs	Converts \t in a string to spaces.
	str_isalnum	Checks whether a string contains only letters and digits.
	str_isalpha	Checks whether a string contains only letters.
	str_isascii	Checks whether a string is in the ASCII table.
	str_isdecimal	Checks whether a string contains only decimal characters.
	str_isdigit	Checks whether a string contains only digits.
	str_isidentifier	Checks whether a string is a valid Python identifier or checks whether a variable name is valid.
Character set check	str_islower	Checks whether a string contains lowercase letters.
	str_isnumeric	Checks whether a string contains digits.

str_isprintable	Checks whether all characters in a string are printable characters.
str_isspace	Checks whether a string contains only spaces.
str_istitle	Checks whether the first letter of each word in a string is in uppercase and the other letters in the string are in lowercase.
str_isupper	Checks whether all letters in a string are in uppercase.

The following table describes the functions that can be used together with string functions.

Category	Function	Description
Multi-string operation	op_add	Returns the sum value among multiple numeric values or strings.
	op_max	Returns the maximum value among multiple numeric values or strings.
	op_min	Returns the minimum value among multiple numeric values or strings.
String truncation	op_slice	Truncates a string.
Length calculation	op_len	Returns the length of a string.

str_format

The str_format function formats strings.

• Syntax

str_format(format_string, value1, value2, ...)

Parameters

Parameter	Туре	Required	Description
format_string	Arbitrary (automatically converted to the string type)	Yes	The format of the output string. Example: {} ={} .
valuel	Arbitrary	Yes	The value that you want to format.
value2	Arbitrary	Yes	The value that you want to format.

Response

- A formatted string is returned.
- Example
- Raw log

class: Format escape_name: Traditional

• Transformation rule

e_set("str_format", str_format("{}={}", v("class"), v("escape_name")))

• Result

class: Format escape_name: Traditional str_format: Format=Traditional

str_join

The str_join function concatenates input strings to generate a new string by using a specified connector.

• Syntax

str_join(connector, value1, value2,)

• Parameters

Parameter	Туре	Required	Description
connector	Arbitrary (automatically converted to the string type)	Yes	The connector. Supported connectors include the exclamation point (!), at sign ($@$), number sign (#), dollar sign (\$), and percent sign (%).
valuel	Arbitrary (automatically converted to the string type)	Yes	The value that you want to concatenate.
value2	Arbitrary (automatically converted to the string type)	Yes	The value that you want to concatenate.

Response

A concatenated string is returned.

- Example
 - Raw log

name: ETL company: aliyun.com

• Transformation rule

e_set("email", str_join("@", v("name"), v("company")))

• Result

name: ETL company: aliyun.com email:ETL@aliyun.com

str_encode

The str_encode function encodes a string by using a specified encoding format.

• Syntax

str_encode(value, "utf8", errors="ignore"))

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The value that you want to encode.
encoding	String	No	The encoding format. Default value: utf8. ASCII is supported.
			The method that is used to process characters if some characters cannot be recognized based on the encoding format. Valid values:
			 ignore (default): No characters are encoded.
errors	String	No	 strict: reports an error and discards the log that contains the characters.
			 replace: replaces the unrecognized characters with question marks (?).
			 xmlcharrefreplace: replaces the unrecognized characters with XML characters.

Response

- An encoded string is returned.
- Examples
- Example 1
 - Raw log
 - test: asewds
 - Transformation rule

e_set("f1", str_decode(str_encode("hello", "utf8"), "utf8"))

Result

test: asewds f1: hello

• Example 2

Raw log

f2: test test data

Transformation rule

e_set("f1", str_encode(v("f2"), "ascii", errors="ignore"))

Result

f1:test f2:test test data

• Example 3

- Raw log
 - f2: test test data

Transformation rule

e_set("f1", str_encode(v("f2"), "ascii", errors="strict"))

Result

An error is reported during execution.

- Example 4
 - Raw log
 - f2: test test dataTransformation rule
 - ----
 - e_set("f1", str_encode(v("f2"), "ascii", errors="replace"))
 - Result

f1:test ???? f2:test test data

- Example 5
- Raw log
 - f2: test test data
- Transformation rule

e_set("f1", str_encode(v("f2"), "ascii", errors="xmlcharrefreplace"))

Result

f1:test 测试数据 f2:test test data

str_decode

The str_decode function decodes an input value by using a specified encoding format.

Syntax

str_decode(value, "utf8", errors="ignore"))

(?) Note The str_decode function can process only the data of the byte data type.

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The value that you want to decode.
encoding	Arbitrary (automatically converted to the string type)	No	The encoding format. Default value: utf8. ASCII is supported.
errors	Arbitrary (automatically converted to the string type)	No	 The method that is used to process characters if some characters cannot be recognized based on the encoding format. Valid values: ignore (default): No characters are decoded. strict: reports an error and discards the log that contains the characters. replace: replaces the unrecognized characters with question marks (?). xmlcharrefreplace: replaces the unrecognized characters with XML characters.

- Response
- A decoded value is returned.
- Example
- Raw log
 - test: asewds
- Transformation rule

e_set("encoding", str_decode(b'\xe4\xbd\xa0\xe5\xa5\xbd', "utf8", 'strict'))

• Result

test: asewds encoding: hello

str_replace

The str_replace function replaces an existing string with a specified string based on a specified rule.

Syntax

str_replace(value, old, new, count)

② Note You can call this function by passing basic variable parameters. For more information, see Function invoking.

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The value in which you want to replace a string.

old	Arbitrary (automatically converted to the string type)	Yes	The string that you want to replace.
new	Arbitrary (automatically converted to the string type)	Yes	The string that you want to use to replace the specified string.
count	Number	No	The number of replacements. If you do not configure this parameter, the specified string in all occurrences in the value is replaced.

Response

A new string is returned.

• Example

Convert a dictionary to the JSON format.

Raw log

content: {'referer': '-', 'request': 'GET /phpMyAdmin', 'status': 404, 'data-1': {'aaa': 'Mozilla', 'bbb': 'asde'}, 'data-2': {'up_adde':
'-', 'up_host': '-'}}

Transformation rule

e_set("content_json", str_replace(ct_str(v("content")),"'",'"'))

• Result

content: {'referer': '-', 'request': 'GET /phpMyAdmin', 'status': 404, 'data-1': {'aaa': 'Mozilla', 'bbb': 'asde'}, 'data-2': {'up_adde':
'-', 'up_host': '-'}}
content_json: {"referer": "-", "request": "GET /phpMyAdmin", "status": 404, "data-1": {"aaa": "Mozilla", "bbb": "asde"}, "data-2":

{"up_adde": "-", "up_host": "-"}}

str_sort

The str_sort function sorts a specified object.

Syntax

str_sort(value, reverse=False)

Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to sort.
reverse	Bool	No	Default value: False. This value indicates that the string is sorted in ascending order.

Response

A sorted string is returned.

- Examples
 - Example 1: Sort the value of the str field in alphabetical order.
 - Raw log

str: twish

Transformation rule

e_set("str_sort", str_sort(v("str")))

Result

str: twish str_sort: histw

• Example 2: Sort the value of the str field in reverse alphabetical order at a granularity of two-letter pairs.

 Raw lo 	g
----------------------------	---

str: twish

Transformation rule

e_set("str_sort", str_sort(v("str"), reverse=True))

Type

Result

str: twish str_sort: wtsih

str_reverse

The str_reverse function reverses a string.

• Syntax

str_reverse(value)

• Parameters

Parameter

Required

Description

	value	Arbitrary (automatically converted to the string type)	Yes	The value that you want to reverse.		
• R	lesponse					
Д	A reversed string is returned.					
• E	• Example					
R	everse the value of the data field.					
0	Raw log					
	data:twish					
0	Transformation rule					
	<pre>e_set("reverse_data", str_reverse</pre>	e(v("data")))				
0	Result					
	data:twish reverse_data:hsiwt					
st	r logstash config normal	lize				
The	e str_logstash_config_normalize funct	tion converts data in the Logstash cor	nfiguration language to the JSON form	at.		
• 5	yntax					
	<pre>str_logstash_config_normalize(value</pre>	2)				
	② Note For more information a	bout the Logstash configuration lange	uage, see <mark>Logstash</mark> .			
• P	arameters					
	Parameter	Туре	Required	Description		
	Value	Type Arbitrary (automatically converted to the string type)	Required Yes	Description The value that you want to convert.		
• R	Parameter value	Type Arbitrary (automatically converted to the string type)	Required Yes	Description The value that you want to convert.		
• R	Parameter value lesponse	Type Arbitrary (automatically converted to the string type)	Required Yes	Description The value that you want to convert.		
• R A	Parameter value desponse a converted string is returned. axample	Type Arbitrary (automatically converted to the string type)	Required Yes	Description The value that you want to convert.		
• R A • E	Parameter value desponse a converted string is returned. ixample convert the value of the field in Logst	Type Arbitrary (automatically converted to the string type)	Required Yes	Description The value that you want to convert.		
• R A • E C	Parameter value esponse converted string is returned. ixample convert the value of the field in Logst Raw log	Type Arbitrary (automatically converted to the string type)	Required Yes	Description The value that you want to convert.		
• R A • E C o	Parameter value tesponse converted string is returned. trample convert the value of the field in Logst Raw log logstash: {"name"=>"tw5"}	Type Arbitrary (automatically converted to the string type)	Required Yes	Description The value that you want to convert.		
• R A • E C o	Parameter value desponse a converted string is returned. axample convert the value of the field in Logst Raw log logstash: ("name"=>"tw5") Transformation rule	Type Arbitrary (automatically converted to the string type)	Required Yes	Description The value that you want to convert.		
• R A • E C o	Parameter value desponse a converted string is returned. axample convert the value of the field in Logst Raw log logstash: ("name"=>"tw5") Transformation rule e_set("normalize_data", str_logst	Type Arbitrary (automatically converted to the string type) ash. ash.	Required Yes	Description The value that you want to convert.		
• F A • E C o	Parameter value value desponse a converted string is returned. axample convert the value of the field in Logst Raw log logstash: ("name"=>"tw5") Transformation rule e_set("normalize_data", str_logst Result	Type Arbitrary (automatically converted to the string type) ash. ash.	Yes	Description The value that you want to convert.		
• R A • E C o	Parameter value va	Type Arbitrary (automatically converted to the string type) assh. assh_config_normalize (v ("logstash"))	Yes	Description The value that you want to convert.		
• R A • E C 0 0	Parameter value value desponse a converted string is returned. example convert the value of the field in Logst Raw log logstash: ("name"=>"tw5") Transformation rule e_set("normalize_data", str_logst Result logstash: ("name"=>"tw5") normalize_data:("name":"tw5")	Type Arbitrary (automatically converted to the string type) ash.	Yes	Description The value that you want to convert.		
• F A • E C o o st	Parameter value desponse a converted string is returned. ixample convert the value of the field in Logst Raw log logstash: ("name"=>"tw5") Transformation rule e_set("normalize_data", str_logst Result logstash: ("name"=>"tw5") r_hex_escape_encode est r hex escape encode function escape	Type Arbitrary (automatically converted to the string type) ash. ash_config_normalize(v("logstash")) capes special characters. The function	Required Yes	Description The value that you want to convert. to Chinese characters.		
• R A • E C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Parameter value tesponse a converted string is returned. ixample convert the value of the field in Logst Raw log logstash: {"name"=>"tw5"} Transformation rule e_set("normalize_data", str_logst Result logstash: {"name"=>"tw5"} r_hex_escape_encode estr_hex_escape_encode function estront estr_hex_escape_encode function estront	Type Arbitrary (automatically converted to the string type) ash. ash_config_normalize (v ("logstash")) capes special characters. The function	Required Yes	Description The value that you want to convert. to Chinese characters.		
• F A • E C • • • • • • • • •	Parameter value value tesponse a converted string is returned. ixample convert the value of the field in Logst Raw log logstash: ("name"=>"tw5") Transformation rule e_set("normalize_data", str_logst Result logstash: ("name"=>"tw5") normalize_data:("name":"tw5") r_hex_escape_encode e str_hex_escape_encode function esc syntax str_hex_escape_encode (value)	Type Arbitrary (automatically converted to the string type) assh. assh_config_normalize (v ("logstash"))	Required Yes	Description The value that you want to convert. to Chinese characters.		
• R A • E C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Parameter value desponse acconverted string is returned. ixample convert the value of the field in Logst Raw log logstash: ("name"=>"tw5") Transformation rule e_set("normalize_data", str_logst Result logstash: ("name"=>"tw5") r_hex_escape_encode e str_hex_escape_encode function escape str_hex_escape_encode(value) arameters	Type Arbitrary (automatically converted to the string type) ash. ash. cash_config_normalize(v("logstash")) capes special characters. The function	Required Yes	Description The value that you want to convert. to Chinese characters.		
• P A • E C • • • • • • • • • • • • •	Parameter value value desponse a converted string is returned. xample convert the value of the field in Logst Raw log logstash: ("name"=>"tw5") Transformation rule e_set("normalize_data", str_logst Result logstash: ("name"=>"tw5") r_hex_escape_encode estr_hex_escape_encode function escontration extr_hex_escape_encode function escontration str_hex_escape_encode (value) arameters Parameter	Type Arbitrary (automatically converted to the string type) ash. ash_config_normalize (v ("logstash")) capes special characters. The function Type	Required Yes n can escape hexadecimal characters Required	Description The value that you want to convert. to Chinese characters. Description		

Response

An escaped string is returned.

Example

Escape the value of the myfriend field to Chinese characters.

Raw log

myfriend: \xE6\x9F\xB3\xE4\xBA\x91

• Transformation rule

e_set("hex_myfriend", str_hex_escape_encode("myfriend"))

• Result

hex_myfriend:myfriend myfriend:\xE6\x9F\xB3\xE4\xBA\x91

str_strip

The str_strip function deletes specified characters from a string.

• Syntax

.

str_strip(value, chars)

• Parameters

Parameter	Туре	Required	Description		
value	Arbitrary (automatically converted to the string type)	Yes	The original string that you want to modify.		
chars	Arbitrary (automatically converted to the string type)	No	The character set that you want to delete from the start and end of the specified string. Default value: $\t r\n$.		
Response					
A modified string is returned.					
Examples					
• Example 1: Delete the asterisks (*) from the start of the value of the strip field.					

Raw log

strip: ***I love Etl

Transformation rule

e_set("str_strip", str_strip(v("strip"), "*"))

```
    Result
```

strip: ***I love Etl
str_strip:I love Etl

$\circ~$ Example 2: Delete the spaces from the start of the value of the strip field.

- Raw log
 - strip: I love Etl

Transformation rule

e_set("str_strip", str_strip(v("strip")))

Result

strip: I love Etl str_strip: I love Etl

$\circ~$ Example 3: Delete the character set ~ $_{\rm xy}$.

Raw log

strip:xy123yx

Transformation rule

e_set("str_strip", str_strip(v("strip"), "xy"))

Result

strip:xy123yx str_strip:123

str_lower

The str_lower function converts all uppercase letters in a string to lowercase letters.

Syntax

str_lower(value)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to convert.

Response

A converted string is returned.

- Example
- Convert the value of the name field to lowercase letters.
- Raw log

name: Etl

Transformation rule

e_set("str_lower", str_lower(v("name")))

• Result

name: Etl str_lower: etl

str_upper

The str_upper function converts all lowercase letters in a string to uppercase letters.

Syntax

str_upper(value)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to convert.

Response

A converted string is returned.

Example

Convert the value of the name field to uppercase letters.

Raw log

name: etl

• Transformation rule

e_set("str_upper", str_upper(v("name"))))

• Result

name: etl str_upper: ETL

str_title

The str_title function capitalizes the first letter of each word in a string and converts the other letters in the string to lowercase letters.

Syntax

str_title(value)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to convert.

- Response
 - A converted string is returned.
- Example

Capitalize the first letter of each word in the value of the word field.

Raw log

word: this is etl

• Transformation rule

e_set("str_title", str_title(v("word")))

Result

word: this is etl str_title: This Is Etl

str_capitalize

The str_capitalize function capitalizes the first letter of a string and converts the other letters in the string to lowercase letters.

Syntax

str_capitalize(value)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to convert.

- Response
- A converted string is returned.
- Example

Capitalize the first letter of the value of the word field and convert the other letters in the value to lowercase letters.

Raw log

word: this Is MY EAL

• Transformation rule

e_set("str_capitalize", str_capitalize(v("word")))

• Result

word: this Is MY EAL str_capitalize: This is my eal

str_lstrip

The str_lstrip function deletes specified characters from the start of a string.

• Syntax

str_lstrip(value, chars)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string that you want to modify.
chars	Arbitrary (automatically converted to the string type)	No	The character set that you want to delete from the start of a string. The default value is a space.

- Response
- A modified string is returned.
- Examples
- $\circ~$ Example 1: Delete the asterisks (*) from the start of the value of the word field.
 - Raw log

word: ***this is string

Transformation rule

e_set("str_lstrip", str_lstrip(v("word"), "*"))

Result

word: ***this is string
str_lstrip: this is string

 $\circ~$ Example 2: Delete the spaces from the start of the value of the word field.

Raw log

word: this is string

Transformation rule

e_set("str_lstrip", str_lstrip(v("word")))

Result

word: this is string
str_lstrip: this is string

- $\circ~$ Example 3: Delete the character set $$_{\rm xy}$$.
- Raw log

lstrip:xy123yx

Transformation rule

e_set("str_lstrip", str_lstrip(v("lstrip"),"xy"))

Result

lstrip:xy123yx str_lstrip:123yx

str_rstrip

The str_rstrip function deletes specified characters from the end of a string.

Syntax

str_rstrip(value, chars)

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string that you want to modify.

User Guide-Log Service

Cloud Defined Storage

	chars	Arbitrary (automatically converted to the string type)	No	The character set that you want to delete from the end of a string. The default value is a space.		
• F / • E 0	Response A modified string is returned. Examples Example 1: Delete the asterisks (*) • Raw log	from the end of the value of the word	l field.			
	word: this is string*****					
	 Transformation rule 					
	e_set("str_rstrip", str_rstrip	(v("word"), "*"))				
	 Result 					
	<pre>word: this is string***** str_rstrip: this is string</pre>					
0	Example 2: Delete the character se Raw log	t xy .				
	word:xy123yx					
	 Transformation rule 					
	<pre>e_set("str_rstrip", str_rstrip</pre>	(v("word"), "xy"))				
	 Result 					
	word:xy123yx str_rstrip:xy123					
ct	r swapcase					
The	e str_swapcase function converts the Syntax	uppercase letters to lowercase letter	s and lowercase letters to uppercase	etters in a string.		
The • 9	<pre>str_swapcase function converts the Syntax str_swapcase(value)</pre>	uppercase letters to lowercase letter	s and lowercase letters to uppercase	etters in a string.		
50 The • 9	str_swapcase function converts the syntax str_swapcase(value) Parameters	uppercase letters to lowercase letter	s and lowercase letters to uppercase	etters in a string.		
• 5	e str_swapcase function converts the syntax str_swapcase (value) Parameters Parameter	uppercase letters to lowercase letter	s and lowercase letters to uppercase Required	etters in a string. Description		
5 The • 9 • F	e str_swapcase function converts the syntax str_swapcase (value) Parameters Parameter value	uppercase letters to lowercase letter Type Arbitrary (automatically converted to the string type)	s and lowercase letters to uppercase Required Yes	etters in a string. Description The string that you want to convert.		
 SL The S F F A A E O 	e str_swapcase function converts the syntax str_swapcase (value) Parameters Parameter value Response A converted string is returned. Example Raw log	Uppercase letters to lowercase letter Type Arbitrary (automatically converted to the string type)	s and lowercase letters to uppercase Required Yes	etters in a string. Description The string that you want to convert.		
• 5 • 5 • F • F • F • E	e str_swapcase function converts the syntax str_swapcase (value) Parameters Parameter value Response A converted string is returned. Example Raw log name: this is string	Type Arbitrary (automatically converted to the string type)	Required Yes	etters in a string. Description The string that you want to convert.		
• 5 • 5 • F • F • F • E • 0	e str_swapcase function converts the syntax str_swapcase (value) Parameters Parameter value Response A converted string is returned. Example Raw log name: this is string Transformation rule	uppercase letters to lowercase letter Type Arbitrary (automatically converted to the string type)	rs and lowercase letters to uppercase Required Yes	etters in a string. Description The string that you want to convert.		
• F • F • F • F • E • O	e str_swapcase function converts the syntax str_swapcase (value) Parameters Parameters value value Response Converted string is returned. Example Raw log name: this is string Transformation rule e_set ("str_swapcase", str_swapcase	Type Arbitrary (automatically converted to the string type)	Required Yes	etters in a string. Description The string that you want to convert.		
• F • F • F • F • C	e str_swapcase function converts the syntax str_swapcase (value) Parameters Parameters value Response Converted string is returned. Example Raw log name: this is string Transformation rule e_set ("str_swapcase", str_swapcase Result	Type Arbitrary (automatically converted to the string type) se (v ("name")))	Required Yes	etters in a string. Description The string that you want to convert.		
 F F F F E 0 0 	e str_swapcase function converts the syntax str_swapcase (value) Parameters Parameters value Response A converted string is returned. Example Raw log name: this is string Transformation rule e_set ("str_swapcase", str_swapcase Result name: this is string str_swapcase: THIS IS STRING	Type Arbitrary (automatically converted to the string type) se (v ("name")))	Required Yes	etters in a string. Description The string that you want to convert.		
• F • F • F • E • C • C • C • C • C	e str_swapcase function converts the Syntax str_swapcase (value) Parameters Parameter value Response A converted string is returned. Example Raw log name: this is string Transformation rule e_set ("str_swapcase", str_swapcas Result name: this is string str_swapcase: THIS IS STRING r translate	uppercase letters to lowercase letter Type Arbitrary (automatically converted to the string type) se (v("name")))	Required Yes	etters in a string. Description The string that you want to convert.		
 The S F F F E O O S S The 	e str_swapcase function converts the syntax str_swapcase (value) Parameters Parameter value Response A converted string is returned. Example Raw log name: this is string Transformation rule e_set ("str_swapcase", str_swapcase Result name: this is string str_swapcase: THIS IS STRING r_translate e str_translate function replaces spece	uppercase letters to lowercase letter Type Arbitrary (automatically converted to the string type) se (v("name"))) ified characters in a string with mapp	Required Yes ing characters.	etters in a string. Description The string that you want to convert.		
 Fractional Sector 10 (1998) Fractional Sector 10 (1998) State 10 (1998) State 10 (1998) 	e str_swapcase function converts the syntax str_swapcase (value) Parameters Parameters Parameter value Response A converted string is returned. Example Raw log name: this is string Transformation rule e_set("str_swapcase", str_swapcase Result name: this is string str_swapcase: THIS IS STRING r_translate e str_translate function replaces spect syntax	uppercase letters to lowercase letter Type Arbitrary (automatically converted to the string type) se (v("name"))) ified characters in a string with mapper	Required Yes	etters in a string. Description The string that you want to convert.		

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string in which you want to replace characters.
replace_string	Arbitrary (automatically converted to the string type)	Yes	The original character set whose characters you want to replace.
mapping_string	Arbitrary (automatically converted to the string type)	Yes	The character set that is used to replace characters in the original character set.

- Response
- A processed string is returned.
- Example

Raw log

name: I love ETL!!!

• Transformation rule

e_set("str_translate", str_translate(v("name"), "aeiou", "12345"))

• Result

name: I love ETL!!!
str_translate: I l4v2 ETL!!!

str_endswith

The str_endswith function checks whether a string ends with a specified suffix.

Syntax

str_endswith(value, suffix, start, end)

③ Note You can call this function by passing basic variable parameters. For more information, see Function invoking.

Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string that you want to check.
suffix	Arbitrary (automatically converted to the string type)	Yes	The suffix. The value of this parameter can be a string or an element.
start	Number	No	The position from which the string suffix check starts. The value 0 indicates the first character. The value -1 indicates the last character.
end	Number	No	The position at which the string suffix check ends. The value 0 indicates the first character. The value -1 indicates the last character.

Response

If the string ends with the specified suffix, the value True is returned. Otherwise, the value False is returned.

- Example
- Raw log

name: this is endswith!!!

• Transformation rule

e_set("str_endswith",str_endswith(v("name"), "!"))

• Result

name: this is endswith!!!
str_endswith: True

str_startswith

The str_starts with function checks whether a string starts with a specified string.

Syntax

str_startswith(value, prefix, start, end)

③ Note You can call this function by passing basic variable parameters. For more information, see Function invoking.

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string that you want to check.
prefix	Arbitrary (automatically converted to the string type)	Yes	The prefix. The value of this parameter can be a string or an element.
start	Number	No	The position from which the string prefix check starts. The value 0 indicates the first character. The value -1 indicates the last character.

			The position at which the string prefix check ends.
end	Number	No	The value 0 indicates the first character. The value -1 indicates the last character.

Response

If the string starts with the specified prefix, the value True is returned. Otherwise, the value False is returned.

Example

• Raw log name: !! this is startwith

Transformation rule

e_set("str_startswith",str_startswith(v("name"), "!!"))

• Result

name: !! this is startwith
str_startswith: True

str_find

The str_find function checks whether a string contains a specified substring.

Syntax

str_find(value, str, begin, end)

③ Note You can call this function by passing basic variable parameters. For more information, see Function invoking.

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string in which you want to search for a substring.
str	Arbitrary (automatically converted to the string type)	Yes	The substring that you want to search for.
begin	Number	No	The position from which the substring search starts. The default value 0 indicates the first character. The value -1 indicates the last character.
end	Number	No	The position at which the substring search ends. The default value is the length of the string. The value 0 indicates the first character. The value -1 indicates the last character.

Response

The position of the specified substring in the original string is returned. If the specified substring appears multiple times in the original string, only the position of the first occurrence of the substring is returned.

- Example
- Raw log

```
name: hello world
```

• Transformation rule

e_set("str_find", str_find(v("name"), "h"))

Result

name: hello world
str_find: 0

str_count

The str_count function counts the number of occurrences of a character in a string.

```
• Syntax
```

str_count(value, sub, start, end)

③ Note You can call this function by passing basic variable parameters. For more information, see Function invoking.

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string in which you want to count the number of occurrences of a character.

sub	Arbitrary (automatically converted to the string type)	Yes	The character whose number of occurrences you want to count.
start	Number	No	The position from which the search for the specified character starts in the string. Valid values: • 0 (default value): the first character. • -1: the last character.
end	Number	No	The position at which the search for the specified character ends in the string. Valid values: • 0: the first character. • -1 (default value): the last character.

Response

The number of occurrences of the specified character is returned.

- Example
 - Raw log

name: this is really a string

Transformation rule

e_set("str_count", str_count(v("name"), "i"))

• Result

name: this is really a string
str_count: 3

str_rfind

The str_rfind function returns the position of the last occurrence of a specified string or a specified character in a string.

• Syntax

str_rfind(value, substr, beg, end)

⑦ Note You can call this function by passing basic variable parameters. For more information, see Function invoking.

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string in which you want to search for a character.
substr	Arbitrary (automatically converted to the string type)	Yes	The character that you want to search for.
beg	Number	No	The position from which the search starts. Default value: 0.
end	Number	No	The position at which the search ends. The default value is the length of the string.

Response

The position of the last occurrence of the specified character or string is returned.

Example
 Raw log

name: this is really a string

• Transformation rule

e_set("str_rfind", str_rfind(v("name"), "i"))

Result

name: this is really a string
str_rfind: 20

str_split

The str_split function splits a string by using a specified delimiter.

• Syntax

str_split(value, sep=None, maxsplit=-1)

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string that you want to split.

sep	Number	No	The delimiter. The value None indicates a space.
maxsplit	Number	No	The maximum number of strings into which you can split the original string. The value -1 indicates no limit.

- Response
 - A processed string is returned.
- Example
- Split the value of the content field by using the space delimiter.
- Raw log
 - content: hello world
- Transformation rule

e_set("str_split", str_split(v("content"), " "))

Result

content: hello world
str_split: ["hello", "world"]

str_splitlines

The str_splitlines function splits a string by using a line feed.

Syntax

str_splitlines(value, keepends)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string that you want to split.
keepends	Bool	No	Specifies whether to delete line feeds from the output result. The line feeds include $\ r$,

- Response
- Processed strings are returned.
- Examples
 - Example 1
 - Raw log

content: ab c\n\nde fg\rkl\r\n

Transformation rule

e_set("str_splitlines", str_splitlines(v("content"), False))

Result

content: ab c\n\nde fg\rkl\r\n
str_splitlines: ['ab c', '', 'de fg', 'kl']

- Example 2
- Raw log

content: ab c\n\nde fg\rkl\r\n

Transformation rule

e_set("str_splitlines", str_splitlines(v("content"), True))

Result

content: ab c\n\nde fg\rkl\r\n
str_splitlines: ['ab c\n', '\n', 'de fg\r', 'kl\r\n']

str partition

The str_partition function splits a string into three parts from left to right by using a specified delimiter.

Syntax

str_partition(value, substr)

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to split.

User Guide-Log Service

	substr	Arbitrary (automatically converted to the string type)	No	The specified delimiter.					
• R S	Response Split strings are returned.								
• E	Example								
S °	Split the value of the website field into three parts from left to right by using the solution. delimiter.								
	website: www.aliyun.com								
0	Transformation rule								
	<pre>e_set("str_partition", str_partit</pre>	ion(v("website"), "."))							
0	Result								
	website: www.aliyun.com str_partition: ["www", ".", "al:	.yun.com"]							
sti The • S	r_rpartition e str_rpartition function splits a string syntax	into three parts from right to left by	using a specified delimiter.						
	<pre>str_rpartition(value, substr)</pre>								
• P	Parameters								
	Parameter	Туре	Required	Description					
	value	Arbitrary (automatically converted to the string type)	Yes	The original string that you want to split.					
	substr	Arbitrary (automatically converted to	No	The specified delimiter					
		the string type)							
• R	Response	the string type)							
• R P	Response Processed strings are returned.	the string type)							
• R P • E	Response Processed strings are returned. Example	the string type)							
• R P • E S	Response Processed strings are returned. Example Split the value of the website field int	the string type)	ng the . delimiter.						
• R P • E S	Response Processed strings are returned. Example Split the value of the website field int Raw log	the string type) o three parts from right to left by usin	ng the delimiter.						
• R P • E S	Response Processed strings are returned. Example Split the value of the website field int Raw log website: www.aliyun.com	the string type) o three parts from right to left by usin	ng the , delimiter.						
• R P • E S o	Response Processed strings are returned. Example Split the value of the website field int Raw log website: www.aliyun.com Transformation rule	the string type) o three parts from right to left by usin	ng the delimiter.						
• R P • E S o	Response Processed strings are returned. Example Split the value of the website field int Raw log website: www.aliyun.com Transformation rule e_set("str_partition", str_rpart:	the string type) three parts from right to left by usin tion (v ("website"), "."))	ng the delimiter.						
• R P • E S o	Response Processed strings are returned. Example Split the value of the website field int Raw log website: www.aliyun.com Transformation rule e_set("str_partition", str_rpart: Result	the string type) to three parts from right to left by usin tion (v ("website"), "."))	ng the delimiter.						
• R P • E S o o	Response Processed strings are returned. Example Split the value of the website field int Raw log website: www.aliyun.com Transformation rule e_set("str_partition", str_rpart: Result website: www.aliyun.com str_partition: ["www.aliyun", '	<pre>the string type) to three parts from right to left by usin tion(v("website"), ".")) '.", "com"]</pre>	ng the , delimiter.						

The str_center function pads a string to a specified length by using a specified character.

Syntax

str_center(value, width, fillchar)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string that you want to modify.
width	Number	Yes	The length of the string after padding.
fillchar	Arbitrary (automatically converted to the string type)	No	The character that is used for padding. The default value is a space.

Response

A processed string is returned.

③ Note If the length of the string after padding is less than the length of the original string, the original string is returned.

• Examples

- $\circ~$ Example 1: Pad a string by using asterisks ($~\star~$).
- Raw log

center: this is center

Transformation rule

e_set("str_center", str_center(v("center"), 40, "*"))

Result

- Example 2: Pad a string by using spaces.
 - Raw log

center: this is center

Transformation rule

e_set("str_center", str_center(v("center"), 40))

Result

center: this is center str_center: this is center

str_zfill

The str_zfill function pads a string to a specified length by using 0 from the start of the string.

• Syntax

str_zfill(value, width)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string that you want to modify.
width	Number	Yes	The length of the string after padding.

- Response
 - A processed string is returned.
- Example
- Raw log

center: this is zfill

• Transformation rule

e_set("str_zfill", str_zfill(v("center"), 40))

• Result

str_expandtabs

The str_expandtabs function converts \times in a string to spaces.

• Syntax

str_expandtabs(value, tabsize)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string that you want to modify.
tabsize	Number	Yes	The number of spaces after conversion.

Response

A processed string is returned.

• Examples

- $\circ~$ Example 1: Convert $~\hfill tensor ten$
- Raw log

logstash: this is\tstring

Transformation rule

e_set("str_expandtabs", str_expandtabs(v("logstash")))

Result

logstash: this is\tstring
str_expandtabs: this is string

- $\circ~$ Example 2: Convert $~\table t$ in the value of the center field to spaces.
- Raw log

{"center": "this is\tstring"}

Transformation rule

e_set("str_expandtabs", str_expandtabs(v("center"), 16))

string

Result

center: this is\tstring str_expandtabs: this is

str_ljust

The str_ljust function pads a string to a specified length by using a specified character from the end of the string.

• Syntax

str_ljust(value, width, fillchar)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string that you want to modify.
width	Number	Yes	The length of the string after padding.
fillchar	Arbitrary (automatically converted to the string type)	No	The character that is used for padding. The default value is a space.

Response

A processed string is returned.

② Note If the length of the string after padding is less than the length of the original string, the original string is returned.

- Examples
 - Example 1
 - Raw log

content: this is ljust

Transformation rule

e_set("str_ljust", str_ljust(v("content"), 20, "*"))

Result

content: this is ljust
str_ljust: this is ljust*******

• Example 2

Raw log

center: this is ljust

Transformation rule

e_set("str_ljust", str_ljust(v("center"), 20,))

Result

center: this is ljust str_ljust: this is ljust

- Example 3: The value of width is less than the length of the original string. The original string is returned.
- Raw log

center: this is ljust

Transformation rule

e_set("str_ljust", str_ljust(v("center"),10, "*"))

Result

center: this is ljust
str_ljust: this is ljust

str_rjust

The str_rjust function pads a string to a specified length by using a specified character from the start of the string.

Syntax

str_rjust(value, width, fillchar)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The original string that you want to modify.
width	Number	Yes	The length of the string after padding.
fillchar	Arbitrary (automatically converted to the string type)	No	The character that is used for padding. The default value is a space.

Response

A processed string is returned.

(?) Note If the length of the string after padding is less than the length of the original string, the original string is returned.

• Example

Raw log

center: this is rjust

• Transformation rule

e_set("str_rjust", str_rjust(v("center"), 20, "*"))

• Result

center: this is rjust str_rjust: ******this is rjust

str_zip

The str_zip function concurrently splits two values or strings that are returned by expressions and combines the results into one string.

• Syntax

str_zip(value1, value2, combine_sep=None, sep=None, quote=None, lparse=None, rparse=None)

Parameter	Туре	Required	Description
value1	Arbitrary (automatically converted to the string type)	Yes	The value that you want to combine.
value2	Arbitrary (automatically converted to the string type)	Yes	The value that you want to combine.
combine_sep	Arbitrary (automatically converted to the string type)	No	The identifier that is used when elements are combined. Default value: # .
sep	Arbitrary (automatically converted to the string type)	No	The delimiter that is used between the elements after the combination. The value must be a single character. Default value:
quote	Arbitrary (automatically converted to the string type)	No	The character is used to enclose the elements after the combination. This parameter is required if the values contain delimiters. Default value:
lparse	Arbitrary (automatically converted to the string type)	No	The delimiter and quote that are used among the elements of value1 . Default delimiter: , . Default quote: " . Format: lparse=(',', '"') . O Note The quote has a higher priority than the delimiter.

	rparse	Arbitrary (automatically	No	The delimiter and quote that are used among the elements of value2 . Default delimiter , . Default quote: ". Format: rparse=(',', '"') .						
		converted to the string type)		⑦ Note The quote has a higher priority than the delimiter.						
•	Response A combined string is returned. Examples • Example 1									
	 Raw log website: wwww.aliyun.co 	m								
	escape_name: o									
Transformation rule										
	 Besult 	.p(v(website), v(escape_nam	e), comprine_sep- e))							
	<pre>website: wwww.aliyun.co escape_name: o combine: wwww.aliyun.co</pre>	meo								
	Example 2Raw log									
	<pre>website: wwww.aliyun.co escape_name: o</pre>	m								
	 Transformation rule 									
	e_set("combine", str_zi	p(v("website"), v("escape_nam	e"))))							
	 Result 									
	<pre>website: wwww.aliyun.co escape_name: o wwww.aliyun.com#o</pre>	un.								
	 Example 3: In this example, Raw log 	, the sep parameter is used.								
	f1: a,b,c f2: x,y,z									
	 Transformation rule 									
	e_set("combine", str_zi	.p(v("f1"), v("f2"), sep=" "))								
	 Result 									
	f1: a,b,c f2: x,y,z combine: a#x b#y c#z									
	 Example 4: In this example, Raw log 	, the quote parameter is used.								
	f1: "a,a", b, "c,c" f2: x, "y,y", z									
	 Transformation rule 									
	e_set("combine", str_zi	.p(v("f1"), v("f2"), quote=' '))							
	 Result 									
	f1: "a,a", b, "c,c" f2: x, "y,y", z combine: a,a#x , b#y,y , c,c#z									
	 Example 5: In this example, Raw log 	, field values with different len	gths are used.							
	f1: a,b f2: x,y,z									
	 Transformation rule 									
	e_set("combine", str_zi	p(v("f1"), v("f2")))								
	 Result 									
	f1: a,b f2: x,y,z combine: a#x,b#y									

- Example 6: In this example, the Iparse and rparse parameters are used.
- Raw log
 - f1: a#b#c f2: x|y|z
- Transformation rule

e_set("combine", str_zip(v("f1"), v("f2"), lparse=("#", '"'), rparse=("|", '"')))

Result

f1: a#b#c
f2: x|y|z
combine: a#x,b#y,c#z

- Example 7: In this example, the Iparse and rparse parameters are used.
 - Raw log

f1: |a,a|, b, |c,c| f2: x, #y,y#, z

Transformation rule

e_set("combine", str_zip(v("f1"), v("f2"), lparse=(",", '|'), rparse=(",", '#')))

Result

f1: |a,a|, b, |c,c| f2: x, #y,y#, z combine: "a,a#x","b#y,y","c,c#z"

str_isalnum

The str_isalnum function checks whether a string contains only letters and digits.

Syntax

str_isalnum(value)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to check.

Response

The value True or False is returned.

Example

• Raw log

- content: 13
- Transformation rule

e_set("str_isalnum", str_isalnum(v("content")))

• Result

content: 13 str_isalnum: True

str_isalpha

The str_isalpha function checks whether a string contains only letters.

Syntax

str_isalpha(value)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to check.

Response

The value True or False is returned.

- Example
 - Raw log
 - content: 13
 - Transformation rule

e_set("str_isalpha", str_isalpha(v("content")))

• Result

content: 13 str_isalpha: False

str_isascii

The str_isascii function checks whether a string is in the ASCII table.

Syntax

str_isascii(value)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to check.

Response

The value True or False is returned.

Example

• Raw log

content: asw123

• Transformation rule

e_set("str_isascii", str_isascii(v("content")))

```
• Result
```

content: asw123 str_isascii: True

str_isdecimal

The str_isdecimal function checks whether a string contains only decimal characters.

Syntax

str_isdecimal(value)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to check.

Response

The value True or False is returned.

- Examples
 - Example 1
 - Raw log

welcome: hello

Transformation rule

e_set("str_isdecimal", str_isdecimal(v("welcome"))))

Result

welcome: hello str_isdecimal: False

- Example 2
 - Raw log

num: 123

Transformation rule

e_set("str_isdecimal", str_isdecimal(v("num")))

Result

num: 123 str_isdecimal: True

str_isdigit

The str_isdigit function checks whether a string contains only digits.

 Syntax

	<pre>str_isdigit(value)</pre>			
•	Parameters			
	Parameter	Туре	Required	Description

User Guide-Log Service

Cloud Defined Storage

value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to check.
Response			
The value True or False is retu	urned.		
Example			
> Raw log			
content: 13			
Transformation rule			
e_set("str_isdigit", str_	isdigit(v("content")))		
Result			
content: 13 str_isdigit: True			

str_isidentifier

The str_isidentifier function checks whether a string is a valid Python identifier or checks whether a variable name is valid.

• Syntax

•

str_isidentifier(value)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to check.
Perpense			

Response

The value True or False is returned.

- Example
- Raw log
 - class: class

• Transformation rule

e_set("str_isidentifier", str_isidentifier(v("class")))

• Result

class: class str_isidentifier: True

str_islower

The str_islower function checks whether a string contains lowercase letters.

Syntax

str_islower(value)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to check.

- Response
- The value True or False is returned.
- Example
- Raw log
- lower: asds
- Transformation rule

e_set("str_islower", str_islower(v("lower")))

• Result

lower: asds str_islower: True

str_isnumeric

The str_isnumeric function checks whether a string contains digits.

• Syntax

str_isnumeric(value)

User Guide-Log Service

	Parameter	Туре	Required	Description
	value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to check.
•	Response The value True or False is returned. Example ° Raw log			
	num: 123			
Transformation rule				
	<pre>e_set("str_isnumeric", str_isnume</pre>	eric(v("num")))		
	• Result			
	num: 123 str_isnumeric: True			
51 다	tr_isprintable ne str_isprintable function checks whe Syntax	ther all characters in a string are print	able characters.	
	<pre>str_isprintable(value)</pre>			
•	Parameters			
	Parameter	Туре	Required	Description
	value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to check.
•	 Response The value True or False is returned. Example Raw log 			
	content: vs;,.as			
	 Transformation rule 			
	<pre>e_set("str_isprintable", str_isp</pre>	printable(v("content"))))		
	• Result			
	<pre>content: vs;,.as str_isprintable: True</pre>			
str_isspace The str_isspace function checks whether a string contains only spaces. • Syntax				
	<pre>str_isspace(value)</pre>			
•	Parameters			
	Parameter	Туре	Required	Description
	value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to check.
•	Response The value True or False is returned.			

- Example
- Raw log

```
space: as afsd
```

• Transformation rule

e_set("str_isspace", str_isspace(v("space")))

• Result

space: as afsd
str_isspace: False

str_istitle

The str_istitle function checks whether the first letter of each word in a string is in uppercase and the other letters in the string are in lowercase.

• Syntax

str_istitle(value)

> Document Version: 20240703

User Guide-Log Service

Cloud Defined Storage

	Parameter	Туре	Required	Description
	value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to check.
• F	Response			
٦	he value True or False is returned.			
• E	Example			
0	Raw log			
	title: Alex Is A Boy			
0	Transformation rule			
	<pre>e_set("str_istitle", str_istitle)</pre>	(v("title")))		
0	Result			
	str_istitle:true title:Alex Is A Boy			
st	r isupper			
The	e str_isupper function checks whethe	r all letters in a string are in uppercas	se.	
• 9	Syntax			
	<pre>str_isupper(value)</pre>			
• F	Parameters			
	Parameter	Туре	Required	Description
	value	Arbitrary (automatically converted to the string type)	Yes	The string that you want to check.
• F	Response			
٦	he value True or False is returned.			
• E	Example			
• Raw log				
	content: ASSD			
0	• Transformation rule			
	<pre>e_set("str_isupper", str_isupper)</pre>	(v("content")))		
0	Result			
	content: ASSD str_isupper: True			
st	r uuid			
The	– e str_uuid function generates a rando	om UUID.		
• 5	Syntax			
	<pre>str_uuid(lower=True)</pre>			
• F	Parameters			
	Parameter	Туре	Required	Description
				Specifies whether the letters in the
	lower	Bool	No	output UUID are in lowercase. Default value: True. This value indicates that the letters are in lowercase.
Response				
A	A UUID is returned.			
• Examples				
• Example 1				
	content. I am Iron man			
	Transformation rule			
	e_set("UUID", str_uuid())			
	 Kesuit 			

content: I am Iron man UUID: e9fcd6b0-b970-11ec-979d-0f7041e65ab8

- Example 2
 - Raw log

content: I am Iron man

Transformation rule

e_set("UUID", str_uuid(lower=False))

Result

content: I am Iron man UUID: 0649211E-B971-11EC-A258-E71F0A2930C5

4.5.8.7.7. Date and time functions

This topic describes the syntax and parameters of date and time functions. This topic also provides examples on how to use date and time functions. All values in log events are stored as strings based on the transformation logic of Log Service domain-specific language (DSL). You must convert data types based on your business requirements.

Date and time functions support the following data types. You can use the functions provided in this topic to convert date and time formats. • String

Example: 2022/07/03 02-41-26.

- UNIX timestamp
- Example: 1559500886.
- Datetime object

Example: 2022-07-01 10:10:10+08:00 and 2022-07-01 10:10:10.

Note A UNIX timestamp is a string.

Among the date and time functions that are described in this topic, only the dt_parse , dt_str , and $dt_parsetimestamp$ functions support the preceding data types. For other functions, you must make sure that the parameter values are of the same type.

Functions

Category	Function	Description	
	dt_parse	Converts a value or the value of a time expression to a datetime object.	
	dt_str	Converts a value or the value of a time expression to a string.	
Common datetime conversion	dt_parsetimestamp	Converts a value or the value of a time expression to a UNIX timestamp.	
	dt_prop	Returns a specific attribute of a value, or returns a specific attribute of the value of a time expression. The attribute can be day or year.	
	dt_now	Returns the current datetime object.	
	dt_today	Returns only the current date.	
Datatima quan	dt_utcnow	Returns the current datetime object in the current time zone.	
Dateume query	dt_fromtimestamp	Converts a UNIX timestamp to a datetime object.	
	dt_utcfromtimestamp	Converts a UNIX timestamp to a datetime object in the current time zone.	
	dt_strptime	Parses a time string into a datetime object.	
	dt_currentstamp	Returns the current UNIX timestamp.	
UNIX timestamp generation	dt_totimestamp	Converts a datetime object to a UNIX timestamp.	
	dt_strftime	Converts a datetime object to a string in a specified format.	
Datetime string generation	dt_strftimestamp	Converts a UNIX timestamp to a string in a specified format.	
dt_truncate		Extracts a time value from a value or the value of a time expression based on a specified time granularity.	
	dt_add	Changes a value or the value of a time expression based on a specified time granularity.	
	dt_MO	Offsets a specified time to the same date of the previous or next Nth Monday. The offset value N is passed to the weekday parameter of the dt_add function.	
	dt_TU	Offsets a specified time to the date of the previous or following Nth Tuesday. The offset value N is passed to the weekday parameter of the dt_add function.	
Datetime change	dt_WE	Offsets a specified time to the date of the previous or following Nth Wednesday. The offset value N is passed to the weekday parameter of the dt_add function.	
	dt_TH	$\begin{array}{llllllllllllllllllllllllllllllllllll$	
	dt_FR	Offsets a specified time to the date of the previous or following Nth Friday. The offset value N is passed to the weekday parameter of the dt_add function.	

dt_SA		$\begin{array}{llllllllllllllllllllllllllllllllllll$
	dt_SU	$\begin{array}{llllllllllllllllllllllllllllllllllll$
Time zone change	dt_astimezone	Converts a value or the value of a time expression to a datetime object in a specified time zone.
Difference generation	dt_diff	Returns the difference between two values or between the values of two time expressions based on a specified time granularity.

dt_parse

The dt_parse function converts a value or the value of a time expression to a datetime object.

• Syntax

dt_parse(value, tz=None)

Parameters

Parameter	Туре	Required	Description
value	String, UNIX timestamp, or datetime object	Yes	The value or time expression.
tz	String	No	The time zone. Default value: None. For more information, see Time zones.

- Response
- A datetime object is returned.
- Examples

$\circ~$ Example 1: Convert the value of the time field to a datetime object.

Raw log

time: 1559500886

Transformation rule

e_set("test_time", dt_parse(v("time")))

Result

time: 1559500886 test_time: 2019-06-02 18:41:26

- Example 2: Convert the value of the time field to a datetime object in the time zone of Shanghai.
- Raw log

time: 2019-06-01 10:10:10 tz: Asia/Shanghai

Transformation rule

e_set("test_time", dt_parse(v("time"),tz=v("tz")))

Result

time: 2019-06-01 10:10:10 tz: Asia/Shanghai test_time: 2019-06-01 10:10:10+08:00

dt_str

The dt_str function converts a value or the value of a time expression to a string.

• Syntax

dt_str(value, fmt="format_string", tz=None)

• Parameters

Parameter	Туре	Required	Description
value	String, UNIX timestamp, or datetime object	Yes	The value or time expression.
fmt	String	No	The format of the string. For more information, seeDate and time formatting directives. By default, the parameter is empty and the format of the string remains unchanged.
tz	String	No	The time zone. Default value: None. For more information, see Time zones.

Response

A time string is returned.

Examples

• Example 1: Convert the value of the time field to a time string of the specified format in the time zone of Tokyo.

Raw log

time: 2019-06-03 02:41:26 fmt: %Y/%m/%d %H-%M-%S tz: Asia/Tokyo

Transformation rule

e_set("dt_str", dt_str(v("time"),fmt=v("fmt"),tz=v("tz")))

Result

time: 2019-06-03 02:41:26 fmt: %Y/%m/%d %H-%M-%S tz: Asia/Tokyo dt_str: 2019/06/03 02-41-26

- Example 2: Convert the value of the time field to a time string of the specified format. In this example, the value of the time field is a UNIX timestamp.
- Raw log

time: 1559500886 fmt: %Y/%m/%d %H-%M-%S

Transformation rule

e_set("dt_str", dt_str(v("time"),fmt=v("fmt")))

Result

time: 1559500886 fmt: %Y/%m/%d %H-%M-%S dt_str: 2019/06/02 18-41-26

 $\circ~$ Example 3: Convert the value of the time field to a time string in the default format.

Raw log

time: 2019-06-03 02:41:26

Transformation rule

e_set("dt_str", dt_str(v("time")))

Result

time: 2019-06-03 02:41:26 dt_str: 2019-06-03 02:41:26

dt_parsetimestamp

The dt_parsetimestamp function converts a value or the value of a time expression to a UNIX timestamp.

• Syntax

dt_parsetimestamp(value, tz=None)

• Parameters

Parameter	Туре	Required	Description
value	String, UNIX timestamp, or datetime object	Yes	The value or time expression.
tz	String	No	The time zone. Default value: None. For more information, see Time zones.

- Response
- A UNIX timestamp is returned.
- Examples

• Example 1: Convert the value of the time field to a UNIX timestamp in the time zone of Tokyo.

Raw log

```
time: 2019-06-03 2:41:26
tz: Asia/Tokyo
```

Transformation rule

e_set("dt_parsetimestamp", dt_parsetimestamp(v("time"),v("tz")))

Result

time: 2019-06-03 2:41:26 tz: Asia/Tokyo dt_parsetimestamp: 1559497286

- $\circ~$ Example 2: Convert the value of the time field to a UNIX timestamp.
- Raw log

time: 2019-06-03 2:41:26

Transformation rule

e_set("dt_parsetimestamp",dt_parsetimestamp(v("time")))

Result

time: 2019-06-03 2:41:26 dt_parsetimestamp: 1559529686

- Example 3: Convert the value of the time field to a UNIX timestamp.
- Raw log

time: 2019-06-03 02:41:26+8:00

Transformation rule

e_set("dt_parsetimestamp",dt_parsetimestamp(v("time")))

Result

time: 2019-06-03 02:41:26+8:00 dt_parsetimestamp: 1559500886

dt_prop

The dt_prop function returns a specific attribute of a value, or returns a specific attribute of the value of a time expression. The attribute can be day or year.

Syntax

dt_prop(value, props)

Parameters

Parameter	Туре	Required	Description
value	String, UNIX timestamp, or datetime object	Yes The value or time expression.	
props	String	Yes	The attribute that you want to obtain. For example, if you set the attribute parameter to year, only the year is returned. Valid Values: day, year, month, hour, second, minute, microsecond, weekday, weekdayname, weekdayshortname, monthname, monthshortname, dayofyear, dayofweek, weekofyear, weekofyear_m, tzname, weekofmonth.

Response

The value of the attribute is returned.

- Examples
 - Example 1: Extract the value of the day attribute from the time field.
 - Raw log

time: 2018-10-2 09:11:40

Transformation rule

e_set("dt_parsetimestamp",dt_prop(dt_parse(v("time")),"day"))

Result

time: 2018-10-2 09:11:40 dt_parsetimestamp: 2

- Example 2: Extract the value of the year attribute from the time field.
- Raw log
 - time: 2018-10-2 09:11:40
- Transformation rule

e_set("dt_parsetimestamp",dt_prop(dt_parse(v("time")),"year"))

Result

time: 2018-10-2 09:11:40 dt_parsetimestamp: 2018

- $\circ~$ Example 3: Extract the value of the weekdayname attribute from the time field.
- Raw log

time: 2018-10-2 09:11:40 weekdayname: weekdayname

Transformation rule

e_set("dt_prop",dt_prop(dt_parse(v("time")),"weekdayname"))

Result

time: 2018-10-2 09:11:40 dt_prop: Tuesday

• Example 4: Extract the value of the weekofyear attribute from the time field.

Raw log

time: 2018-10-2 09:11:40

Transformation rule

e_set("dt_prop",dt_prop(dt_parse(v("time")),"weekofyear"))

Result

time: 2018-10-2 09:11:40 dt_prop: 39

dt_now

The dt_now function returns the current datetime object.

- Syntax
 - dt_now(tz=None)

Parameters

Parameter	Туре	Required	Description
tz	String	No	The time zone. Default value: None. For more information, see Time zones.

Response

A datetime object in the specified time zone is returned.

• Example

Return the current datetime object in the time zone of Shanghai.

Raw log

- tz: Asia/Shanghai
- Transformation rule

e_set("dt_now",dt_now(tz=v("tz")))

Result

tz: Asia/Shanghai dt_now: 2022-06-30 11:21:25.111836+08:00

dt_today

The dt_today function returns only the current date.

Syntax

dt_today(tz=None)

• Parameters

Parameter	Туре	Required	Description
tz	String	No	The time zone. Default value: None. For more information, see Time zones.

Response

A date object in the specified time zone is returned.

- Example
 - Return only the current date.

Raw log

- None
- Transformation rule

e_set("dt_today", dt_today())

• Result

dt_today: 2022-06-30 00:00:00

dt_utcnow

The dt_utcnow function returns the current datetime object in the current time zone.

- Syntax
- dt_utcnow()
- Parameters
- None.
- Response
- The current datetime object in the current time zone is returned.
- Example
 - Return the current datetime object in the current time zone.
- Raw log
 - None

Transformation rule

e_set("dt_utcnow",dt_utcnow())

Result

dt_utcnow:2022-06-30 03:33:56.614005

dt_fromtimestamp

The dt_fromtimestamp function converts a UNIX timestamp to a datetime object.

• Syntax

dt_fromtimestamp(value, tz=None)

Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value or time expression.
tz	String	No	The time zone. Default value: None. For more information, see Time zones.

Response

A datetime object is returned.

- Examples
- $\circ\;$ Example 1: Convert the value of the time field to a datetime object.
- Raw log
 - time: 1559500886
- Transformation rule

e_set("dt_fromtimestamp",dt_fromtimestamp(v("time")))

Result

time: 1559500886 dt_fromtimestamp: 2019-06-02 18:41:26

- Example 2: Convert the value of the time field to a datetime object in the time zone of Shanghai.
- Raw log
 - time: 1559500886 tz: Asia/Shanghai
- Transformation rule

e_set("dt_fromtimestamp",dt_fromtimestamp(v("time"),tz=v("tz")))

Result

time: 1559500886 tz: Asia/Shanghai dt_fromtimestamp: 2019-06-03 02:41:26+08:00

dt_utcfromtimestamp

The dt_utcfromtimestamp function converts a UNIX timestamp to a datetime object in the current time zone.

• Syntax

dt_utcfromtimestamp(value)

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value or time expression.

Response

A datetime object is returned.

- Example
- Raw log

time: 1559500886

• Transformation rule

e_set("dt_utcfromtimestamp",dt_utcfromtimestamp(v("time")))

• Result

time: 1559500886 dt_utcfromtimestamp: 2019-06-02 18:41:26

dt_strptime

The dt_strptime function parses a time string into a datetime object.

• Syntax

dt_strptime(value, "format_string")

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value or time expression.
fmt	String	No	The format of the string. For more information, see Date and time formatting directives.

- Response
- A datetime object is returned.
- Example
- Raw log

time: 2019/06/03 02-41-26 fmt: %Y/%m/%d %H-%M-%S

• Transformation rule

e_set("dt_strptime", dt_strptime(v("time"), v("fmt")))

```
• Result
```

time: 2019/06/03 02-41-26 fmt: %Y/%m/%d %H-%M-%S dt_strptime: 2019-06-03 02:41:26

dt_currentstamp

The dt_currentstamp function returns the current UNIX timestamp.

• Syntax

dt_currentstamp(value, normalize='floor')

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value or time expression.
normalize	String	No	 The numeric format in which you want the function to return the result. Valid values: floor (default): rounds a number down to the nearest integer. int : returns the integer part of a number. round : rounds a number to the nearest integer. ceil : rounds a number up to the nearest integer.

Response

The current UNIX timestamp is returned.

Example

Raw log

- None
- Transformation rule

e_set("dt_currentstamp",dt_currentstamp())

```
• Result
```

dt_currentstamp: 1656560437

dt_totimestamp

The dt_totimestamp function converts a datetime object to a UNIX timestamp.

• Syntax

dt_totimestamp(timeexpression)

• Parameters

Parameter	Туре	Required	Description
timeexpression	Datetime object	Yes	The datetime object that you want to convert.

- Response
- A UNIX timestamp is returned.
- Example
- Raw log
- time: 2019-06-03 2:41:26

• Transformation rule

e_set("dt_totimestamp",dt_totimestamp(dt_parse(v("time"))))

• Result

time: 2019-06-03 2:41:26 dt_totimestamp: 1559529686

dt_strftime

The dt_strftime function converts a datetime object to a string in a specified format.

• Syntax

dt_strftime(timeexpression, "format_string")

• Parameters

Parameter	Туре	Required	Description
timeexpression	Datetime object	Yes	The datetime object that you want to convert.
format_string	String	Yes	The format of the string. For more information, see Date and time formatting directives.

Response

A formatted string is returned.

• Example

- Convert the value of the time field to a string of the specified format.
- Raw log

time: 2019-06-03 2:41:26 fmt: %Y/%m/%d %H-%M-%S

• Transformation rule

e_set("dt_strftime",dt_strftime(dt_parse(v("time")),v("fmt")))

• Result

time: 2019-06-03 2:41:26 fmt: %Y/%m/%d %H-%M-%S dt_strftime: 2019/06/03 02-41-26

dt_strftimestamp

The dt_strftimestamp function converts a UNIX timestamp to a string in a specified format.

Syntax

dt_strftimestamp(value, fmt="format_string", tz=None)

Parameters

Parameter	Туре	Required	Description
value	String	Yes	The UNIX timestamp that you want to convert.
fmt	String	Yes	The format of the string. For more information, see Date and time formatting directives.
tz	String	No	The time zone. Default value: None. For more information, see Time zones.

Response

- A formatted string is returned.
- Examples

- Example 1
 - Raw log

time: 1559500886 fmt: %Y/%m/%d %H-%M-%S

Transformation rule

e_set("dt_strftimestamp",dt_strftimestamp(v("time"),v("fmt")))

Result

time: 1559500886 fmt: %Y/%m/%d %H-%M-%S dt_strftimestamp: 2019/06/02 18-41-26

• Example 2

Raw log

time: 1559500886 fmt: %Y/%m/%d %H-%M-%S tz: Asia/Tokyo

Transformation rule

 $\texttt{e_set("dt_strftimestamp",dt_strftimestamp(v("time"),v("fmt"),v("tz")))}$

Result

dt_strftimestamp:2019/06/03 03-41-26
fmt:%Y/%m/%d %H-%M-%S
time:1559500886
tz:Asia/Tokyo

dt_truncate

The dt_truncate function extracts a time value from a value or the value of a time expression based on a specified time granularity.

• Syntax

dt_truncate(value, unit='day')

Parameters

Parameter	Туре	Required	Description	
value	String, UNIX timestamp, or datetime object	Yes	The value or time expression.	
unit	String	Yes	The time granularity that you want to obtain. Default value: day. Valid values: second , minute , <num>_minute (for example, 5_minute, 19_minute, and 2_minute), hour , day , week , month , quarter , half_year , and year .</num>	

Response

The extracted time value is returned.

- Examples
 - Example 1
 - Raw log

time: 2019-06-03 2:41:26 unit: year

Transformation rule

e_set("dt_truncate", dt_truncate(v("time"),v("unit")))

Result

time: 2019-06-03 2:41:26 unit: year dt_truncate: 2019-01-01 00:00:00

• Example 2

Raw log

time: 2019-06-03 2:41:26 unit: hour

Transformation rule

e_set("dt_truncate",dt_truncate(v("time"),v("unit")))

Result

time: 2019-06-03 2:41:26 unit: hour dt_truncate: 2019-06-03 02:00:00

dt_add
The dt_add function changes a value or the value of a time expression based on a specified time granularity.

• Syntax

dt add(value, dtl=None, dt2=None, year(s)=None, month(s)=None, day(s)=None, hour(s)=None, minute(s)=None, second(s)=None, microsecond(s)=None, week(s)=None, weekday=None)

• Parameters

Parameter	Туре	Required	Description
value	String, UNIX timestamp, or datetime object	Yes	The datetime expression.
dtl	String, UNIX timestamp, or datetime object	No	The datetime expression. Default value: None.
dt2	String, UNIX timestamp, or datetime object	No	The datetime expression. Default value: None.
year/years	Number	No	 year: the year that is used to replace the year in the specified time. For example, you can set year=2020. Default value: None. years: the number of years by which you want to offset the specified time. For example, if you set years=1, the function increases the year by one year.
day/days	Number	No	 day: the day that is used to replace the day in the specified time. For example, you can set day=1. Default value: None. days: the number of days by which you want to offset the specified time. For example, if you set days=1, the function increases the day by one day.
hour/hours	Number	No	 hour: the hour that is used to replace the hour in the specified time. For example, you can set hour=1. Default value: None. hours: the number of hours by which you want to offset the specified time. For example, if you set hours=1, the function increases the hour by one hour.
minute/minutes	Number	No	 minute: the minute that is used to replace the minute in the specified time. For example, you can set minute=1. Default value: None. minutes: the number of minutes by which you want to offset the specified time. For example, if you set minutes=1, the function increases the minute by one minute.
second/seconds	Number	No	 second: the second that is used to replace the second in the specified time. For example, you can set second=1. Default value: None. seconds: the number of seconds by which you want to offset the specified time. For example, if you set seconds=1, the function increases the second by one second.
microsecond/microseconds	Number	No	 microsecond: the microsecond that is used to replace the microsecond in the specified time. For example, you can set microsecond=1 Default value: None. microseconds: the number of microseconds by which you want to offset the specified time. For example, if you set microseconds=1 the function increases the microsecond by one microsecond.
week/weeks	Number	No	 week: the week that is used to replace the week in the specified time. For example, you can set week=1. Default value: None. weeks: the number of weeks by which you want to offset the specified time. For example, if you set weeks=1, the function increases the week by one week.
weekday	Number	No	The weekday that is used to replace the weekday in the specified time. For example, you can set $\begin{tabular}{lllllllllllllllllllllllllllllllllll$

Response

The new value of the time expression is returned.

• Examples

Example 1Raw log

```
dt: 2018-10-10 1:2:3
dt1: 2018-11-3 11:12:13
dt2: 2018-10-1 10:10:10
```

Transformation rule

e_set("dt_add",dt_add(dt_parse(v("dt")), dt1=dt_parse(v("dt1")), dt2=dt_parse(v("dt2"))))

Result

```
dt:2018-10-10 1:2:3
dt1:2018-11-3 11:12:13
dt2:2018-10-1 10:10:10
dt_add:2018-11-12 02:04:06
```

• Example 2

Raw log

dt: 2018-10-11 02:03:04 year: 2019

Transformation rule

e_set("dt_add", dt_add(dt_parse(v("dt")), year=ct_int(v("year"))))

Result

dt:2018-10-11 02:03:04 dt_add:2019-10-11 02:03:04 year:2019

dt_MO

The dt_MO function offsets a specified time to the same date of the previous or next Nth Monday. The offset value N is passed to the weekday parameter of the dt_add function.

• Syntax

dt_MO(Integer_or_negative)

• Parameters

Parameter	Туре	Required	Description
Integer_or_negative	Number	Yes	The offset value. If you want to pass a negative integer, use op_neg(Positive integer) . For example, use op_neg(1) to indicate -1.

Response

The time that is offset is returned.

- Example
- Raw log

time: 2019-08-13 02:03:04

• Transformation rule

e_set("dt_MO",dt_add(v("time"),weekday=dt_MO(1)))

• Result

time: 2019-08-13 02:03:04 dt_MO: 2019-08-19 02:03:04

dt_TU

The dt_TU function offsets a specified time to the date of the previous or following Nth Tuesday. The offset value N is passed to the weekday parameter of the dt_add function.

• Syntax

dt_TU(Integer_or_negative)

• Parameters

Parameter	Туре	Required	Description
Integer_or_negative	Number	Yes	The offset value. If you want to pass a negative integer, use op_neg(Positive integer) . For example, use op_neg(1) to indicate -1.

Response

The time that is offset is returned.

Examples

For more information about examples, see dt_MO.

dt_WE

The dt_WE function offsets a specified time to the date of the previous or following Nth Wednesday. The offset value N is passed to the weekday parameter of the dt_add function.

Syntax

dt WE(Integer or negative)
---------------------------	---

• Parameters

Parameter	Туре	Required	Description
Integer_or_negative	Number	Yes	The offset value. If you want to pass a negative integer, use op_neg(Positive integer) . For example, use op_neg(1) to indicate -1.

Response

The time that is offset is returned.

- Examples
- For more information about examples, see dt_MO.

dt_TH

The dt_TH function offsets a specified time to the date of the previous or following Nth Thursday. The offset value N is passed to the weekday parameter of the dt_add function.

• Syntax

dt_TH(Integer_or_negative)

• Parameters

Parameter	Туре	Required	Description
Integer_or_negative	Number	Yes	The offset value. If you want to pass a negative integer, use op_neg(Positive integer) . For example, use op_neg(1) to indicate -1.

Response

The time that is offset is returned.

Examples

For more information about examples, see dt_MO.

dt_FR

The dt_FR function offsets a specified time to the date of the previous or following Nth Friday. The offset value N is passed to the weekday parameter of the dt_add function.

Syntax

dt_FR(Integer_or_negative)

• Parameters

Parameter	Туре	Required	Description
Integer_or_negative	Number	Yes	The offset value. If you want to pass a negative integer, use op_neg(Positive integer) . For example, use op_neg(1) to indicate -1.

- Response
- The time that is offset is returned.
- Examples

For more information about examples, see dt_MO.

dt_SA

The dt_SA function offsets a specified time to the date of the previous or following Nth Saturday. The offset value N is passed to the weekday parameter of the dt_add function.

• Syntax

dt_SA(Integer_or_negative)

• Parameters

Parameter	Туре	Required	Description
Integer_or_negative	Number	Yes	The offset value. If you want to pass a negative integer, use op_neg(Positive integer) . For example, use op_neg(1) to indicate -1.

Response

The time that is offset is returned.

- Examples
- For more information about examples, see dt_MO.

dt_SU

The dt_SU function offsets a specified time to the date of the previous or following Nth Sunday. The offset value N is passed to the weekday parameter of the dt_add function.

Syntax

dt_SU(Integer_or_negative)

• Parameters

Parameter	Туре	Required	Description
Integer_or_negative	Number	Yes	The offset value. If you want to pass a negative integer, use op_neg(Positive integer) . For example, use op_neg(1) to indicate -1.

Response

The time that is offset is returned.

• Examples

For more information about examples, see dt_MO.

dt_astimezone

The dt_astimezone function converts a value or the value of a time expression to a datetime object in a specified time zone.

Syntax

dt_astimezone(value, tz, reset=False)

• Parameters

Parameter	Туре	Required	Description
value	String, UNIX timestamp, or datetime object	Yes	The value or time expression.
tz	String	No	The time zone. Default value: None. For more information, see Time zones.
reset	Bool	No	Specifies whether to change the time zone. The default value False specifies that the datetime object is returned in the current time zone. The value True specifies that the datetime object is returned in the specified time zone.

Response

A datetime object in the specified time zone is returned.

- Examples
 - Example 1
 - Raw log

time: 2019-06-03 2:41:26 tz: UTC

Transformation rule

e_set("dt_astimezone", dt_astimezone(dt_parse(v("time")), v("tz")))

Result

time: 2019-06-03 2:41:26 tz: UTC dt_astimezone: 2019-06-03 02:41:26+00:00

• Example 2

Raw log

time: 2019-06-01 10:10:10+10:00 tz: Asia/Tokyo

Transformation rule

e_set("dt_astimezone", dt_astimezone(v("time"), v("tz"), reset=True))

Result

time: 2019-06-01 10:10:10+10:00 tz: Asia/Tokyo dt_astimezone: 2019-06-01 10:10:10+09:00

- Example 3
 - Raw log

time: 2019-06-01 10:10:10+08:00 tz: Asia/Tokyo

Transformation rule

```
e_set("dt_astimezone", dt_astimezone(v("time"), v("tz"), reset=False))
e_set("dt_astimezone_true", dt_astimezone(v("time"), v("tz"), reset=True))
```

Result

dt_astimezone:2019-06-01 11:10:10+09:00 dt_astimezone_true:2019-06-01 10:10:10+09:00 time:2019-06-01 10:10:10+08:00 tz:Asia/Tokyo

dt_diff

The dt_diff function returns the difference between two values or between the values of two time expressions based on a specified time granularity.

Syntax

dt_diff(value1, value2, unit='second', normalize='floor')

Parameters

Parameter	Туре	Required	Description
valuel	String, UNIX timestamp, or datetime object	Yes	The value or time expression.
value2	String, UNIX timestamp, or datetime object	Yes	The value or time expression.
unit	String	No	The time granularity in which you want the function to return the difference. Default value: <pre>second</pre> . Valid values: second, microsecond , millisecond , minutes , hours , and <pre>day</pre> .
normalize	String	No	 The numeric format in which you want the function to return the result. Valid values: floor (default): rounds a number down to the nearest integer. int : returns the integer part of a number. round : retains N decimal places. ceil : rounds a number up to the nearest integer.

Response

The difference between two values is returned based on the specified time granularity.

Examples

• Example 1: Calculate the difference between the value of the time1 field and the value of the time2 field. Unit: seconds.

Raw log

time1: 2018-10-1 10:10:10 time2: 2018-10-1 10:10:10

Transformation rule

e_set("diff",dt_diff(v("time1"), v("time2")))

Result

time1: 2018-10-1 10:10:10 time2: 2018-10-1 10:10:10 diff: 0

• Example 2: Calculate the difference between the value of the time1 field and the value of the time2 field. Unit: seconds.

Raw log

```
time1: 2018-10-1 11:10:10
time2: 2018-10-1 10:10:10
```

```
    Transformation rule
```

e_set("diff",dt_diff(v("time1"), v("time2")))

Result

time1: 2018-10-1 11:10:10 time2: 2018-10-1 10:10:10 diff: 3600 • Example 3: Calculate the difference between the value of the time1 field and the value of the time2 field. Unit: microseconds.

Raw log

time1: 2018-10-1 11:10:11 time2: 2018-10-1 10:10:10 unit: microsecond

Transformation rule

e_set("diff",dt_diff(v("time1"), v("time2"),v("unit")))

Result

diff:3601000000 time1:2018-10-1 11:10:11 time2:2018-10-1 10:10:10 unit:microsecond

• Example 4: Calculate the difference between the value of the time1 field and the value of the time2 field and round the return value down to the nearest integer. Unit: minutes.

```
    Raw log
```

time1: 2018-10-1 11:11:59 time2: 2018-10-1 10:10:00 unit: minute normalize: floor

Transformation rule

e_set("diff", dt_diff(v("time1"), v("time2"), v("unit"), v("normalize")))

Result

diff:61 normalize:floor time1:2018-10-1 11:11:59 time2:2018-10-1 10:10:00 unit:minute

• Example 5: Calculate the difference between the value of the time1 field and the value of the time2 field and round the return value down to the nearest integer. Unit: seconds.

```
    Raw log
```

time1: 10:00:00
time2: 11:00:00
unit: second
normalize: floor

Transformation rule

e_set("diff", dt_diff(v("time1"), v("time2"), v("unit"), v("normalize")))

Result

```
diff:-3600
normalize:floor
time1:10:00:00
time2:11:00:00
unit:second
```

4.5.8.7.8. Regular expression functions

This topic describes the syntax and parameters of regular expression functions. This topic also provides examples on how to use the functions.

Functions

Category	Function	Description
Value extraction	regex_select	Extracts a value that matches a regular expression.
Value extraction	regex_findall	Extracts all values that match a regular expression.
Evaluation	regex_match	Checks whether a value matches a regular expression.
Replacement	regex_replace	Replaces the characters that match a regular expression in a string.
Splitting	regex_split	Splits a string into an array of strings.

regex_select

The regex_select function extracts a value that matches a regular expression.

Syntax

regex_select(value, r"regular expression", mi=None, gi=None)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary	Yes	The input value.

regular expression	String	Yes	The regular expression.
mi	int	No	The sequence number of the value that is matched in the input value and is returned. Default values: None and 0. The default values indicate that the first value that is matched is returned.
gi	int	No	The sequence number of the group that is used for matching in the regular expression. Default values: None and 0. The default values indicate that the first group is used.

Response

The extracted value is returned.

• Examples

• Example 1: Extract the first value that matches the regular expression from the str field.

Raw log

str: iZbp1a65x3r1vhpe94fi2qZ

Transformation rule

e_set("regex", regex_select(v("str"), r"\d+")) e_set('regex2", regex_select(v('str'), r'\d+", mi=None))
e_set("regex3", regex_select(v("str"), r"\d+", mi=0))

Result

regex:1 regex2:1 regex3:1 str:iZbpla65x3r1vhpe94fi2qZ

• Example 2: Extract the first and second values that match the regular expression from the str field.

Raw log

str: abc123 xyz456

Transformation rule

Extract the first value that matches the regular expression from the str field. e_set("regex", regex_select(v("str"), r"\d+")) # Extract the second value that matches the regular expression from the str field.

- e_set("regex2", regex_select(v("str"), r"\d+", mi=1))
- Result

regex: 123 regex2: 456 str: abc123 xyz456

• Example 3

Raw log

str: abc123 xyz456

Transformation rule

Extract the first value that matches the first group in the regular expression from the str field. e_set("regex", regex_select(v("str"),r"[a-z]+(\d+)",gi=0))
Extract the second value that matches the first group in the regular expression from the str field.

e_set("regex2", regex_select(v("str"),r"[a-z]+(\d+)",mi=1,gi=0)) # Extract the first value that matches the first group in the regular expression from the str field.

e_set("regex3", regex_select(v("str"),r"([a-z]+)(\d+)",gi=0))

Extract the first value that matches the second group in the regular expression from the str field. e_set("regex4", regex_select(v("str"),r"([a-z]+)(\d+)",gi=1))

Result

str: abc123 xyz456 regex: 123 regex2: 456 regex3: abc regex4: 123

regex_findall

The regex_findall function extracts all values that match a regular expression.

Syntax

regex_findall(value, r"regular expression")

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary	Yes	The input value.
regular expression	String	Yes	The regular expression.

Response

The values that match the regular expression are returned.

Cloud Defined Storage

• Example

Extract all numbers from the str field.

• Raw log

str: iZbp1a65x3r1vhpe94fi2qZ

• Transformation rule

e_set("regex_findall", regex_findall(v("str"),r"\d+"))

• Result

str: iZbpla65x3r1vhpe94fi2qZ
regex_findall: ["1", "65", "3", "1", "94", "2"]

regex_match

The regex_match function checks whether a value matches a regular expression.

• Syntax

regex_match(value, r"regular expression", full=False)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary	Yes	The input value.
regular expression	String	Yes	The regular expression.
full	Bool	No	Specifies whether to perform exact match. Default value: False.

Response

The value True or the value False is returned.

Example

Check whether the str field contains numbers.

Raw log

str: iZbp1a65x3r1vhpe94fi2qZ

• Transformation rule

Check whether the str field contains numbers.

- e_set("regex_match", regex_match(v("str"),r"\d+"))
 # Check whether the str field contains only numbers.
- # Check whether the str field contains only numbers. e_set("regex_match2", regex_match(v("str"),r"\d+",full=True))
- Result

str: iZbpla65x3rlvhpe94fi2qZ
regex_match: True
regex_match2: False

regex_replace

The regex_replace function replaces the characters that match a regular expression in a string.

• Syntax

regex_replace(value, r"regular expression", replace="", count=0)

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary	Yes	The input value whose characters you want to replace.
regular expression	String	Yes	The regular expression.
replace	String	No	The characters that you want to use to replace the matched characters in the input value. This parameter is empty by default, which indicates that the matched characters are deleted. You can specify a regular expression. Example: r*(1****\2" . This value indicates that the output value must match the regular expression. • \1 indicates the first group. • \2 indicates the second group.
count	Number	No	The number of times that you want to replace the matched characters. Default value: 0. This value indicates that all matched characters are replaced.

Response

The value after replacement is returned.

Examples

- Example 1: Replace all numbers in the str field with 13.
- Raw log

str: iZbp1a65x3r1vhpe94fi2qZ replace: 13

Transformation rule

e_set("regex_replace", regex_replace(v("str"),r"\d+",v("replace")))

Result

str: iZbp1a65x3r1vhpe94fi2qZ replace: 13 regex_replace: iZbp13a13x13r13vhpe13fi13qZ

• Example 2: Mask four digits in the middle of a mobile phone number.

Raw log

iphone: 13900001234

Transformation rule

e_set(

"sec iphone",

regex_replace(v("iphone"), r"(\d{0,3})\d{4}(\d{4})", replace=r"\1****\2"),)

? Note

- replace=r"\1****\2" indicates that the value after replacement must match the regular expression r"\1****\2".
- 1 indicates the first group. In this example, $(d{0,3})$ is the first group.
- 12 indicates the second group. In this example, $(1d{4})$ is the second group.

Result

iphone: 13900001234 sec_iphone: 139****1234

regex_split

The regex_split function splits a string into an array of strings.

Syntax

regex_split(value, r"regular expression", maxsplit=0)

Parameters

Parameter	Туре	Required	Description
value	Arbitrary	Yes	The input value that you want to split.
regular expression	String	Yes	The regular expression.
maxsplit	int	No	The maximum number of times that the input value can be split. Default value: 0. This value indicates that the input value is split based on all matched characters. The value 1 indicates that the input value is split based on only the first matched character.

Response

An array that contains the values after splitting is returned.

Example

Split the str field by number.

Raw log

str: iZbp1a65x3r1vhpe94fi2gZ

Transformation rule

e_set("regex_split", regex_split(v("str"),r"\d+"))

```
    Result
```

str: iZbp1a65x3r1vhpe94fi2qZ regex_split: ["iZbp", "a", "x", "r", "vhpe", "fi", "qZ"]

4.5.8.7.9. Grok function

This topic describes the syntax and parameters of the Grok function. This topic also provides examples on how to use the function.

Description

Regular expression functions are complicated. We recommend that you use the Grok function instead of regular expression functions. For more information, see Regular expression functions. You can use the Grok function together with regular expression functions. Examples:

e_match("content", grok(r"\w+: (%{IP})")) # The Grok pattern matches the abc: 192.0.2.0 or xyz: 192.0.2.2 pattern of a log. e_match("content", grok(r"\w+: (%{IP})", escape=True)) # The Grok pattern does not match the abc: 192.0.2.0 pattern of log data but matches th e \w+: 192.0.2.0 pattern of log data.

The Grok function extracts specified values based on a regular expression.

Syntax

grok(pattern, escape=False, extend=None)

Grok syntax

%{SYNTAX}

% {SYNTAX:NAME }

In the Grok syntax, SYNTAX specifies a predefined regular expression, and NAME specifies a group. Examples:

Equivalent to r"(?:\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\"
Equivalent to r"(?P<source_id>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"
Equivalent to r"(\d{1,3}\.\d{1,3}\.\d{1,3})" "%{IP}" "%{IP:source_id}" ("%{IP}")

The Grok function supports the following grouping modes: Capturing group mode

Some Grok patterns support named capturing groups. You can use only the %{SYNTAX} syntax for these Grok patterns.

"%{SYSLOGBASE}" "%{COMMONAPACHELOG}" "%{COMBINEDAPACHELOG}" "%{HTTPD20_ERRORLOG}" "%{HTTPD24_ERRORLOG}" "%{HTTPD_ERRORLOG}"

Non-capturing group mode Some Grok patterns support non-capturing groups. Examples:

- "%{INT}" "%{YEAR}" "%{HOUR}"

• Parameters

Parameter	Туре	Required	Description
pattern	String	Yes	The Grok syntax.
escape	Bool	No	Specifies whether to escape special characters that are included in regular expressions in non-Grok patterns. Default value: False.
extend	Dict	No	The custom Grok expression.

Examples

• Example 1: Extract the date and reference content.

Raw log

content: 2019 June 24 "I am iron man"

• Transformation rule

e_regex('content',grok('%{YEAR:year} %{MONTH:month} %{MONTHDAY:day} %{QUOTEDSTRING:motto}'))

Result

content: 2019 June 24 "I am iron man" year: 2019 month: June day: 24 motto: "I am iron man"

• Example 2: Extract an HTTP request log.

Raw log

content: 10.0.0.0 GET /index.html 15824 0.043

Transformation rule

e_regex('content',grok('%{IP:client} %{WORD:method} %{URIFATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration}'))

• Result

content: 10.0.0.0 GET /index.html 15824 0.043 client: 10.0.0.0 method: GET request: /index.html bytes: 15824 duration: 0.043

• Example 3: Extract an Apache log.

Raw log

content: 127.0.0.1 - - [13/Apr/2015:17:22:03 +0800] "GET /router.php HTTP/1.1" 404 285 "-" "curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/ 7.19.7 NSS/3.15.3 zlib/1.2.3 libidn/1.18 libssh2/1.4.2"

Transformation rule

e_regex('content',grok('%{COMBINEDAPACHELOG}'))

• Result

content: 127.0.0.1 - - [13/Apr/2015:17:22:03 +0800] "GET /router.php HTTP/1.1" 404 285 "-" "curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/
7.19.7 NS5/3.15.3 zlib/1.2.3 libidn/1.18 libssh2/1.4.2"
clientip: 127.00.1
ident: auth: timestamp: 13/Apr/2015:17:22:03 +0800
verb: GET
request: /router.php
httpversion: 1.1
response: 404
bytes: 285
referrer: "-"
agent: "curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.15.3 zlib/1.2.3 libidn/1.18 libssh2/1.4.2"

• Example 4: Extract a log in the default syslog format.

Raw log

content: May 29 16:37:11 sadness logger: hello world

• Transformation rule

e_regex('content',grok('%{SYSLOGBASE} %{DATA:message}'))

Result

content: May 29 16:37:11 sadness logger: hello world timestamp: May 29 16:37:11 logsource: sadness program: logger message: hello world

• Example 5: Escape special characters.

Raw log

content: Nov 1 21:14:23 scorn kernel: pid 84558 (expect), uid 30206: exited on signal 3

• Transformation rule

e_regex('content',grok(r'%{SYSLOGBASE} pid %{NUMBER:pid} \(%{WORD:program}\), uid %{NUMBER:uid): exited on signal %{NUMBER:signal}'))

The transformation rule contains special characters parentheses (), which are included in regular expressions. If you do not want to escape the parentheses (), set the escape parameter to True. Example:

e_regex('content',grok('%{SYSLOGBASE} pid %{NUMBER:pid} (%{WORD:program}), uid %{NUMBER:uid}: exited on signal %{NUMBER:signal}', escape=T rue))

• Result

content: Nov 1 21:14:23 scorn kernel: pid 84558 (expect), uid 30206: exited on signal 3
timestamp: Nov 1 21:14:23
logsource: scorn
program: expect
pid: 84558
uid: 30206
signal: 3

• Example 6: Extract a log by using a custom Grok expression.

Raw log

content: Beijing-1104,gary 25 "never quit"

• Transformation rule

e_regex('content',grok('%{ID:user_id},%{WORD:name} %{INT:age} %{QUOTEDSTRING:motto}',extend={'ID': '%{WORD}-%{INT}'}))

• Result

content: Beijing-1104,gary 25 "never quit"
user_id: Beijing-1104
name: gary
age: 25
motto: "never quit"

• Example 7: Match JSON data.

Raw log

content: 2019-10-29 16:41:39,218 - INFO: owt.AudioFrameConstructor - McsStats:
{"event":"mediaStats", "connectionId":"31578616547393100", "durationMs":"5000", "ttpPackets":"250", "rtpBytes":"36945", "nackPackets":"0", "nackPackets":"0", "rtpIntervalAvg":"20", "rtpIntervalAvg":"20", "rtpIntervalAvg":"20", "rtpIntervalAvg":"20", "rtpIntervalAvg":"04," "topSendPackets":"1", "rtcpIntervalAvg":"200", "frameIntervalAvg":"104", "rtpIntervalAvg":"04," "frameIntervalAvg":"04," "frameIntervalAvg":"104", "frameIntervalAvg":"104", "frameIntervalAvg":"0", "frameIntervalAvg":"104", "frameIntervalAvg":"04," "frameIntervalAv

Transformation rule

ransionnation rule

• Result

content: 2019-10-29 16:41:39,218 - INFO: owt.AudioFrameConstructor - McsStats:

{"event":"mediaStats","connectionId":"331578616547393100","durationMs":"5000","rtpPackets":"250","rtpBytes":"36945","nackPackets":"0","nackB :"0","rtpIntervalAvg":"20","rtpIntervalMax":"104","rtpIntervalVar":"4","rtcpRecvPackets":"0","rtcpRecvBytes":"0","rtcpSendPackets":"1","rtcp ytes":"32","frame":"250","frameBytes":"36945","timeStampOutOfOrder":"0","frameIntervalAvg":"20","frameIntervalMax":"104","frameIntervalVar":"timeStampIntervalAvg":"20","frameIntervalMax":"104","frameIntervalVar":"timeStampIntervalAvg":"0")

{"event":"mediaStats","connectionId":"331578616547393100","durationMs":"5000","rtpPackets":"250","rtpBytes":"36945","nackPackets":"0","nackPackets":"0","rtpIntervalAvg":"2

• Example 8: Parse a log in the World Wide Web Consortium (W3C) format.

Raw log

json:

content: 2018-12-26 00:00:00 W3SVC2 application001 192.168.0.0 HEAD / - 8000 - 10.0.0.0 HTTP/1.0 - - - - 404 0 64 0 19 0

Transformation rule

Fields that are not supported by the W3C format are displayed as hyphens (-). Therefore, hyphens (-) are used in Grok patterns to match the fields.

e_regex("content",grok('%{DATE:data} %{TIME:time} %{WORD:s_sitename} %{WORD:s_computername} %{IP:s_ip} %{WORD:cs_method} %
{NOTSPACE:cs_uri_stem} - %{NUMBER:s_port} - %{IP:c_ip} %{NOTSPACE:cs_version} - - - - %{NUMBER:sc_status} %{NUMBER:sc_substatus} %{NUMBER:sc_substatus} %{NUMBER:sc_bytes} %{NUMBER:cs_bytes} %{NUMBER:time_taken}'))

Result

```
content: 2018-12-26 00:00:00 W3SVC2 application001 192.168.0.0 HEAD / - 8000 - 10.0.0.0 HTTP/1.0 - - - 404 0 64 0 19 0
data: 18-12-26
time: 00:00:00
s_sitename: W3SVC2
s_computername: application001
s_ip: 192.168.0.0
cs_method: HEAD
cs_uri_stem: /
s_port: 8000
c_ip: 10.0.0.0
cs version: HTTP/1.0
sc_status: 404
sc_substatus: 0
sc_win32_status: 64
sc bytes: 0
cs bytes: 19
time taken: 0
```

4.5.8.7.10. Structured data functions

This topic describes the syntax and parameters of structured data functions. Structured data includes JSON data and XML data. This topic also provides examples on how to use the functions.

Functions

Category	Function	Description
ISON	json_select	Extracts or calculates specific values from a JSON expression by using JMESPath.
1301	json_parse	Parses a value into a JSON object.
XML	xml_to_json	Converts XML data to JSON data.

json_select

The json_select function extracts or calculates specific values from a JSON expression by using JMESPath.

• Syntax

json_select(value, jmes, default=None, restrict=False)

• Parameters

Parameter	Туре	Required	Description
value	String or JSON	Yes	The input JSON expression or field.
jmes	String	Yes	The JMESPath expression. The expression specifies the field whose value is extracted.
default	String	No	If the specified field does not exist, the value of this parameter is returned. Default value: None. This value indicates that no fields are returned.
restrict	Bool	No	 Specifies whether to enable the restricted mode if the value of the specified field is in an invalid JSON format. Default value: False. Valid values: False: The invalid format issue is ignored, and the system continues to transform data. The value of the <i>default</i> parameter is returned. True: The invalid format issue is reported, and the system stops transforming data. The log is discarded.

User Guide-Log Service

Response

The extracted value is returned.

- Examples
- Example 1: Extract the value of the name field from the content field.
- Raw log

content: {"name": "xiaoming", "age": 10}

Transformation rule

e_set("json_filter",json_select(v("content"), "name"))

Result

content: {"name": "xiaoming", "age": 10}
json_filter: xiaoming

• Example 2: Extract all values of the name element from the content field.

Raw log

content: {"name": ["xiaoming", "xiaowang", "xiaoli"], "age": 10}

Transformation rule

e_set("json_filter", json_select(v("content"), "name[*]"))

Result

content: {"name": ["xiaoming", "xiaowang", "xiaoli"], "age": 10}
json_filter: ["xiaoming", "xiaowang", "xiaoli"]

• Example 3: Extract the value of the name3 field from the content field. If the name3 field does not exist, the value of the default parameter is returned.

Raw log

content: {"name": "xiaoming", "age": 10}

Transformation rule

e_set("json_filter", json_select(v("content"), "name3", default="None"))

Result

content: {"name": "xiaoming", "age": 10}
json_filter: None

• Example 4: Extract the value of the name-test element from the content field.

Raw log

content: {"name": {"name-test":"xiaoming"}, "age": 10}

Transformation rule

e_set("json_filter", json_select(v("content"), 'name."name-test"', default=None))

Result

content: {"name": {"name-test":"xiaoming"}, "age": 10}
json_filter: xiaoming

• Example 5: Extract the value of the name-test element from the content field. If the name-test element does not exist, no fields are returned.

Raw log

content: {"name": {"name.test":"xiaoming"}, "age": 10}

Transformation rule

e_set("json_filter", json_select(v("content"), 'name."name-test"', default=None))

Result

content: {"name": {"name-test":"xiaoming"}, "age": 10}

json_parse

The json_parse function parses a value into a JSON object.

• Syntax

json_parse(value, default=None, restrict=False)

Parameters

Parameter	Туре	Required	Description
value	String	Yes	The input field.
default	String	No	If the specified field does not exist, the value of this parameter is returned. Default value: None. This value indicates that no fields are returned.

restrict	Bool	No	 Specifies whether to enable the restricted mode if the value of the specified field is in an invalid JSON format. Default value: False. Valid values: False: The invalid format issue is ignored, and the system continues to transform data. The value of the <i>default</i> parameter is returned. True: The invalid format issue is reported, and the system stops transforming data. The log is discarded. 			
Bosnonso						
A ISON object is returned	L					
Examples						
 Example 1: Extract the Raw log 	JSON value of the content field.					
content: {"abc":	123, "xyz": "test" }					
 Transformation rule 						
e_set("json", json	_parse(v("content")))					
 Result 						
content: {"abc": ; json: {"abc": 123	123, "xyz": "test" } , "xyz": "test"}					
 Example 2: Extract the Raw log 	value of the content field. If the	value is not in the JSON format,	the value of the default parameter is returned.			
content: this is no	ot json					
 Transformation rule 						
e_set("json", json	_parse(v("content"), default="H	FF", restrict=False))				
 Result 						
content: this is no json: FFF	pt json					
vml to ison						
The xml to ison function c	onverts XML data to ISON data.					
• Syntax						
xml to json(source)						
Parameters						
Parameter	Type	Required	Description			
r urumeter	Chring	Ves	The input field			
source	String	Tes	The input field.			
Response						
JSON-formatted data is re	eturned.					
• Example						
str : <data><country ghbor name="Switzerla ame="Malaysia" direc ica" direction="W"/></country </data>	<pre>name="Liechtenstein"><rank>1<!-- and" direction="W"/-->< tion="N"/><country <neighbor="" direction"<="" na="" name="Colombia" pre=""></country></rank></pre>	rank> <year>2008</year> <gdppc>14: country name="Singapore"><rank> me="Panama"><rank>68</rank><yea: ction="E"/></yea: </rank></gdppc>	1100 <neighbor direction="E" name="Austria"></neighbor> <nei 4<year>2011</year><gdppc>59900</gdppc><neighbor n<br="">r>2011<gdppc>13600</gdppc><neighbor name="Costa R</td></tr><tr><td>• Transformation rule</td><td></td><td></td><td></td></tr><tr><td>e_set(" str_json",="" td="" xml<=""><td>_to_json(v("str")))</td><td></td><td></td></neighbor></neighbor></nei 	_to_json(v("str")))		

• Result

str:<data><country name="Liechtenstein"><rank>1</rank><year>2008</year><gdppc>141100</gdppc><neighbor name="Austria" direction="E"/>
<neighbor name="Switzerland" direction="W"/></country><country name="Singapore"><rank>4</rank><year>2011</year><gdppc>59900</gdppc>
<neighbor name="Malaysia" direction="N"/></country><country name="Panama"><rank>68</rank><year>2011</year><gdppc>13600</gdppc><neighbor name="Colambia" direction="B"/></country></data>
str_json:{"data": {"country": [{"@name": "Liechtenstein", "rank": "1", "year": "2008", "gdppc": "141100", "neighbor": [{"@name": "Switzerland", "@direction": "N"}]}, {"@name": "Singapore", "rank": "68", "year": "2011", "gdppc": "13600",
"neighbor": [{"@name": "Costa Rica", "@direction": "N"}}, {"@name": "Colombia", "@direction": "E"}]}}}

4.5.8.7.11. IP address parsing functions

This topic describes the syntax and parameters of IP address parsing functions. This topic also provides examples on how to use the functions.

Functions

Function	Description
geo_parse	Identifies the city, province, and country based on an IP address.

ip_cidrmatch	Checks whether an IP address belongs to a Classless Inter-Domain Routing (CIDR) block.
ip_version	Checks whether the version of an IP address is IPv4 or IPv6.
ip_type	Identifies the type of an IP address and checks whether the type of the IP address is private or public.
ip_makenet	Converts an IP address to a CIDR block.
ip_to_format	Converts the format of a CIDR block to a format that specifies the netmask or prefix length of the CIDR block.
ip_overlaps	Checks whether two CIDR blocks overlap.
ip2long	Converts an IP address to a value of the long type.
long2ip	Converts a value of the long type to an IP address.

geo_parse

The geo_parse function identifies the city, province, and country based on an IP address.

Syntax

geo_parse(ip, ip_db="SLS-GeoIP", keep_fields=None, provider="ipip", ip_sep=None)

• Parameters

Parameter	Туре	Required	Description
ip	String	Yes	The IP address that you want to parse to obtain the city, province, and country to which the IP address belongs. If you want to enter multiple IP addresses, you can specify the delimiter by using the <i>ip_sep</i> parameter.
ip_db	String	Yes	The IP address database that is used to parse an IP address into the city, province, and country to which the IP address belongs. Valid values: SLS-GeoIP (default): the built-in IP address database of Log Service. To ensure accuracy, the built-in IP address database of Log Service is updated once a day. You can use the database without the need for additional configuration. Custom IP address database: Set the value to res_oss_file(endpoint, ak_id, ak_key, bucket, file, format='binary', change_detect_interval=0, fetch_interval=2, refresh_retr y_max=60, encoding='utf8', error='ignore'). For more information about the parameters in the res_oss_file function, see res_oss_file.
keep_fields	Tuple	Νο	 The keys that are included in the response. If you use the built-in IP address database to parse an IP address, the following keys are included in the response by default: city: the name of the city province: the name of the province country: the name of the country city_en: the administrative region code or name of the city province_en: the administrative region code or name of the city province: the code or name of the country or region isp: the name of the Internet service provider (ISP) lat: the latitude of the location to which the IP address belongs lor: the longitude of the location to which the IP address belongs If you use a custom IP address database to parse an IP address, the following keys are included in the response by default: city: the name of the cuty province: the name of the province country: the name of the country For example, keep_fields=("city", "country") indicates that the city and country keys are returned. The keep_fields parameter can also be used to rename the keys. For example, (("city", "cty"), ("country", "state")) indicates that the city and country keys are renamed cty and state in the returned result.
provider	String	No	 This parameter is valid only when the <i>ip_db</i> parameter is set to a <i>custom IP address database</i>. Valid values: ipip (default): The binary IP address database that is provided by IPIP in the IPDB format is used to parse an IP address. To download the database, visit ipip. ip2location: The global binary IP address database that is provided by IP2Location is used to parse an IP address. To download the database, visit ip2location. Only a binary IP address database is supported.

ip_sep	String	No	The IP address delimiter. The delimiter is used to delimit a string of IP addresses into multiple IP addresses. The response is in the JSON format. Default value: None. This value indicates that IP addresses are not delimited.

Response

A dictionary is returned in the following format:

```
{
  "city": "...",
  "province":"...",
  "country": "..."
}
```

• Examples

- Example 1: Use the built-in IP address database of Log Service to query data
 - Raw log
 - ip : 203.0.113.1
 - Transformation rule

e_set("geo", geo_parse(v("ip")))

- Result
 - ip : 203.0.113.1

geo: {"city":"Hangzhou","province":"Zhejiang province","country":"China","isp":"China Mobile","lat":30.16,"lon":120.12}

- Example 2: Use the built-in IP address database of Log Service to query data. The function parses a log field that contains multiple IP addresses and
 returns the city, province, and country to which each IP address belongs.
 - Raw log

ip : 203.0.113.4, 192.0.2.2, 198.51.100.2

Transformation rule

e_set("geo", geo_parse(v("ip"), ip_sep=","))

Result

```
ip : 203.0.113.4, 192.0.2.2, 198.51.100.2
```

geo : {"203.0.113.4": {"country_en": "CN", "province_en": "330000", "city_en": "330200", "country": "China", "province": "Zhejiang province", "city": "Ningbo", "isp": "China Telecom", "lat": 29.8782, "lon": 121.549}, "192.0.2.2": {"country_en": "CN", "province_en": " 320000", "city_en": "321300", "country": "China", "province": "Jiangsu province", "city": "Sugian", "isp": "China Telecom", "lat": 33.94 92, "lon": 118.296}, "198.51.100.2": {"country_en": "CN", "province_en": "330000", "city_en": "330500", "country": "China", "province": "Zhejiang province", "city": "Huzhou", "isp": "China Telecom", "lat": 30.8703, "lon": 120.093}}

- Example 3: Use a custom IP address database to query data.
 - Raw log

ip : 203.0.113.1

Transformation rule

bucket='your bucket', file='ipipfree.ipdb',

format='binary', change_detect_interval=20)))

Result

```
ip: 203.0.113.1
```

geo : {"city": "Hangzhou", "province":"Zhejiang province","country": "China"}

- Example 4: Use a custom IP address database to query data. The function returns the specified keys and renames the keys.
- Raw log

ip : 203.0.113.1

Transformation rule

```
e_set("geo",geo_parse(v("ip"), ip_db=res_oss_file(endpoint='http://oss-cn-hangzhou.aliyuncs.com',
```

ak_id='your ak_id',

ak_key='your ak_key',

ntry", "state"), ("province", "pro"))))

```
    Result
```

ip : 203.0.113.1

geo : { "state": "China","pro": "Zhejiang province","cty": "Hangzhou"}

- Example 5: Use a custom IP address database to query data. The function returns the specified keys.
- Raw log
 - ip : 203.0.113.1

Transformation rule

e_set("geo",geo_parse(v("ip"), ip_db=res_oss_file(endpoint='http://oss-cn-hangzhou.aliyuncs.com',

ak_id='your ak_id',
ak_key='your ak_key',

bucket='your bucket', file='ipipfree.ipdb',

format='binary', change_detect_interval=20), keep_fields=

("country", "province")))

Result

ip : 203.0.113.1
geo : { "country": "China","province": "Zhejiang province"}

 Example 6: Use a custom IP address database to query data and use the global binary IP address database that is provided by IP2Location to parse the data. The function returns the specified keys.

Raw log

ip : 203.0.113.2

Transformation rule

e_set("geo", geo_parse(v("ip"), ip_db=res_oss_file(endpoint='http://oss-cn-hangzhou.aliyuncs.com',ak_id="your ak_id", ak_key="your ak_serect", bucket='log-etl-staging', file='your ip2location bin file', format='binary', change detect interval=20),provider="ip2location"))

Result

ip : 203.0.113.2 geo : {"city":"Dearborn","province":"Michigan","country":"United States"}

If you set the value of the provider parameter to ip2location, the open source SDK for Python that is provided by IP2Location is used for data transformation. The SDK for Python that is provided by IP2Location can be used to parse the following fields. If a field fails to be parsed, you must check whether the field is included in the IP address database provided by IP2Location.

country_short country_long / The country field is specified for data transformation. region / The province field is specified for data transformation. city isp latitude longitude domain zipcode timezone netspeed idd code area_code weather_code weather_name mcc mnc mobile_brand elevation usage_type

For more information, visit IP2Location Python SDK.

- Example 7: Use a custom IP address database to query data. The function parses a log field that contains multiple IP addresses and returns the city, province, and country to which each IP address belongs.
 - Raw log

ip :	203.0.1	13.3, 1	92.0.2.1,	198.51.100.1
------	---------	---------	-----------	--------------

Transformation rule

- ak_key="ak_serect",
- bucket='log-etl-staging',

file='calendar.csv/IP2LOCATION-LITE-DB3.BIN',
format='binary', change_detect_interval=20),

provider="ip2location", ip_sep=","))

Result

ip : 203.0.113.3, 192.0.2.1, 198.51.100.1

geo : {"203.0.113.3": {"city": "Dearborn", "province": "Michigan", "country": "United States"}, "192.0.2.1": {"city": "Hangzhou", "provi nce": "Zhejiang", "country": "China"}, "198.51.100.1": {"city": "Hangzhou", "province": "Zhejiang", "country": "China"}}

ip_cidrmatch

The ip_cidrmatch function returns a Boolean value based on whether an IP address matches a specified CIDR subnet. This function checks whether an IP address belongs to a CIDR block. If the IP address belongs to the CIDR block, the function returns true. Otherwise, the function returns false. IPv4 and IPv6 addresses are supported.

• Syntax

ip_cidrmatch(cidr_subnet, ip, default="")

Parameters

Parameter	Туре	Required	Description
cidr_subnet	String	Yes	The CIDR block. Example: 192.168.1.0/24.
ip	String	Yes	The IP address.
default	String	No	If the IP address does not belong to the CIDR block, the value of this parameter is returned. You can leave this parameter empty.

Response

If the specified IP address belongs to the specified CIDR block, the function returns true. Otherwise, the function returns false.

• Examples

• Example 1: The specified IPv4 address belongs to the specified CIDR block. The function returns true.

Raw log

cidr_subnet: 192.168.1.0/24 ip: 192.168.1.100

Transformation rule

e_set("is_belong", ip_cidrmatch(v("cidr_subnet"), v("ip")))

Result

```
cidr_subnet: 192.168.1.0/24
ip: 192.168.1.100
is_belong: true
```

• Example 2: The specified IPv4 address does not belong to the specified CIDR block. The function returns false.

Raw log

cidr_subnet: 192.168.1.0/24 ip: 10.10.1.100

Transformation rule

e_set("is_belong", ip_cidrmatch(v("cidr_subnet"), v("ip")))

Result

cidr_subnet: 192.168.1.0/24 ip: 10.10.1.100 is belong: false

• Example 3: The function cannot determine whether the specified IP address belongs to the specified CIDR block and returns unknown.

Raw log

cidr_subnet: 192.168.1.0/24 ip: a

Transformation rule

e_set("is_belong",ip_cidrmatch(v("cidr_subnet"),v("ip"),default="unknown"))

Result

cidr_subnet: 192.168.1.0/24
ip: a
is_belong: unknown

ip_version

The ip_version function checks whether the version of an IP address is IPv4 or IPv6. If an IP address is an IPv4 address, IPv4 is returned. If an IP address is an IPv6 address, IPv6 is returned.

• Syntax

ip_version(ip, default="")

• Parameters

Parameter	Туре	Required	Description
ip	String	Yes	The IP address.
default	String	No	If the version of the specified IP address fails to be identified, the value of this parameter is returned. You can leave this parameter empty.

Response

IPv6 or IPv4 is returned.

Examples

- Example 1: The specified IP address is an IPv4 address. The function returns IPv4.
- Raw log
 - ip: 192.168.1.100
- Transformation rule

e_set("version", ip_version(v("ip")))

Result

ip: 192.168.1.100 version: IPv4

- Example 2: The specified IP address is an IPv6 address. The function returns IPv6.
 - Raw log
 - ip: ::1
 - Transformation rule

e_set("version", ip_version(v("ip")))

Result

ip: ::1 version: IPv6

ip_type

The ip_type function identifies the type of an IP address and checks whether the type of the IP address is private or public. Valid values: private, reserved, loopback, public, and allocated ripe ncc.

Syntax

ip_type(ip, default="")

• Parameters

Parameter	Туре	Required	Description
ip	String	Yes	The IP address.
default	String	No	If the type of the specified IP address fails to be identified, the value of this parameter is returned. You can leave this parameter empty.

Response

A value that indicates the type of an IP address is returned. Valid values: private, reserved, loopback, public, and allocated ripe ncc.

- Examples
- Example 1: Identify the type of the specified IP address. The function returns loopback.
 - Raw log
 - ip: 127.0.0.1
 - Transformation rule

e_set("type", ip_type(v("ip")))

Result

ip: 127.0.0.1 type: loopback

- Example 2: Identify the type of the specified IP address. The function returns private.
 - Raw log
 - ip: 47.100.XX.XX
 - Transformation rule

e_set("type",ip_type(v("ip")))

Result

ip: 47.100.XX.XX

type: private

 $\circ~$ Example 3: Identify the type of the specified IP address. The function returns public.

Raw log

ip: 47.100.XX.XX

Transformation rule

e_set("type", ip_type(v("ip")))

Result

ip: 47.100.XX.XX type: public

Cloud Defined Storage

- Example 4: Identify the type of the specified IPv6 address. The function returns loopback.
 - Raw log
 - ip: ::1
 - Transformation rule

e_set("type", ip_type(v("ip")))

Result

ip: ::1 type: loopback

- $\circ~$ Example 5: Identify the type of the specified IPv6 address. The function returns allocated ripe ncc.
- Raw log

ip: 2001:0658:022a:cafe:0200::1

Transformation rule

e_set("type", ip_type(v("ip")))

Result

ip: 2001:0658:022a:cafe:0200::1
type: allocated ripe ncc

ip_makenet

The ip_makenet function converts an IP address to a CIDR block.

Syntax

ip_makenet(ip, subnet_mask=None, default="")

• Parameters

Parameter	Туре	Required	Description
ip	String	Yes	The IP address.
subnet_mask	String	Yes	The subnet mask. Example: 255.255.255.0. Note If you set the ip parameter to an IP address range, you can leave the subnet_mask parameter empty.
default	String	No	If the specified IP address fails to be converted to a CIDR block, the value of this parameter is returned. You can leave this parameter empty.

- Response
 - A CIDR block is returned.
- Examples
 - $\circ~$ Example 1: Convert an IP address to a CIDR block.
 - Raw log
 - ip: 192.168.1.0
 - Transformation rule

e_set("makenet",ip_makenet(v("ip"),"255.255.255.0"))

Result

ip: 192.168.1.0 makenet: 192.168.1.0/24

- Example 2: Convert an IP address range to a CIDR block.
- Raw log

ip: 192.168.1.0-192.168.1.255

Transformation rule

e_set("makenet", ip_makenet(v("ip")))

Result

ip: 192.168.1.0-192.168.1.255 makenet: 192.168.1.0/24

- Example 3: Convert an IP address range to a CIDR block.
- Raw log

ip: 192.168.1.0/255.255.255.0

Transformation rule

e_set("makenet",ip_makenet(v("ip")))

Result

ip: 192.168.1.0/255.255.255.0 makenet: 192.168.1.0/24

ip_to_format

The ip_to_format function converts the format of a CIDR block to a format that specifies the netmask or prefix length of the CIDR block.

Syntax

ip_to_format(cidr_subnet, want_prefix_len=0, default="")

Parameters

Parameter	Туре	Required	Description
cidr_subnet	String	Yes	The CIDR block. Example: 192.168.1.0/24.
want_prefix_len	Int	No	 The format of the output CIDR block. Default value: 0. Valid values: 0: returns the original CIDR block. 1: returns an IP address and the prefix length of the IP address. 2: returns an IP address and the netmask of the IP address. 3: returns an IP address range.
default	String	No	If the format of the specified CIDR block fails to be converted to the specified format, the value of this parameter is returned. You can leave this parameter empty.

- Response
- A CIDR block of the specified format is returned.
- Examples
 - Example 1: Return the original CIDR block.
 - Raw log

ip: 192.168.1.0/24

Transformation rule

e_set("strNormal",ip_to_format(v("ip"),0))

Result

ip: 192.168.1.0/24
strNormal: 192.168.1.0/24

- Example 2: Convert the format of a CIDR block to the format that specifies the prefix length of the CIDR block.
- Raw log

ip: 192.168.1.0/24

Transformation rule

e_set("strNormal",ip_to_format(v("ip"),1))

Result

ip: 192.168.1.0/24
strNormal: 192.168.1.0/24

• Example 3: Convert the format of a CIDR block to the format that specifies the netmask of the CIDR block.

- Raw log
 - ip: 192.168.1.0/24
- Transformation rule

e_set("strNormal",ip_to_format(v("ip"),2))

Result

ip: 192.168.1.0/24 strNormal: 192.168.1.0/255.255.255.0

- Example 4: Convert a CIDR block to an IP address range.
- Raw log

ip: 192.168.1.0/24

Transformation rule

e_set("strNormal",ip_to_format(v("ip"),3))

Result

ip: 192.168.1.0/24 strNormal: 192.168.1.0-192.168.1.255

ip_overlaps

The ip_overlaps function checks whether two CIDR blocks overlap.

• Syntax

ip_overlaps(cidr_subnet, cidr_subnet2, default="")

• Parameters

Parameter	Туре	Required	Description
cidr_subnet	String	Yes	The first CIDR block.
cidr_subnet2	String	Yes	The second CIDR block.
default	String	No	If the function cannot determine whether the CIDR blocks overlap, the value of this parameter is returned. You can leave this parameter empty.

- Response
 - If the specified CIDR blocks do not overlap, the function returns 0.
 - If the specified CIDR blocks overlap at the end of the blocks, the function returns 1.
 - $\circ~$ If the specified CIDR blocks overlap at the start of the blocks, the function returns -1.
- Examples
 - Example 1: The specified two CIDR blocks do not overlap.
 - Raw log cidr1: 192.168.0.0/23

cidr2: 192.168.2.0/24

Transformation rule

e_set("overlaps", ip_overlaps(v("cidr1"),v("cidr2")))

Result

cidr1: 192.168.0.0/23 cidr2: 192.168.2.0/24 overlaps: 0

• Example 2: The specified two CIDR blocks overlap at the start of the blocks.

Raw log

cidr1: 192.168.1.0/24 cidr2: 192.168.0.0/23

Transformation rule

e_set("overlaps", ip_overlaps(v("cidr1"), v("cidr2")))

Result

```
cidr1: 192.168.1.0/24
cidr2: 192.168.0.0/23
overlaps: -1
```

• Example 3: The specified two CIDR blocks overlap at the end of the blocks.

Raw log

cidr1: 192.168.0.0/23 cidr2: 192.168.1.0/24

Transformation rule

e_set("overlaps", ip_overlaps(v("cidr1"),v("cidr2")))

Result

cidr1: 192.168.0.0/23 cidr2: 192.168.1.0/24 overlaps: 1

ip2long

The ip2long function converts an IP address to a value of the long type.

Syntax

User Guide-Log Service

ip2long(value,default=0)

Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value that you want to convert.
default	String	No	The value that is converted from an invalid IP address. You can use a custom value. Example: 0.

Response

The value that is converted from a valid IP address is returned. The value is of the long type.

• Examples

• Example 1: Convert a valid IP address. This is the default scenario.

•	Raw log
	ip: 192.168.0.100
•	Transformation rule
	<pre>e_set("long_ip",ip2long(v("ip")))</pre>
•	Result
	ip: 192.168.0.100 long_ip: 167772160
∘ E> ∎	xample 2: Convert an invalid IP address. Raw log
	ip: 47.100.XX.XX
•	Transformation rule
	<pre>e_set("long_ip",ip2long(v("ip"), "ignore"))</pre>
	Result

ip:47.100.XX.XX

long_ip:ignore

long2ip

The long2ip function converts a value of the long type to an IP address.

Syntax

long2ip(value,default="")

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value that you want to convert.
default	String	No	The empty string that is converted from an invalid value of the long type. You can use a custom string.

Response

The IP address that is converted from a valid value of the long type is returned.

• Examples

- $\circ~$ Example 1: Convert a valid value of the long type. This is the default scenario.
 - Raw log
 - long: 167772160
 - Transformation rule

e_set("ip",long2ip(v("long")))

Result

long: 167772160

- ip: 192.168.0.100
- Example 2: Convert an invalid value of the long type.
 - Raw log

long: 4294967296

Transformation rule

e_set("ip",long2ip(v("long")))

Result

long: 4294967296

ip:

- Example 3: Convert an invalid value of the long type and set the default parameter to a custom string.
- Raw log
 - long: 4294967296
- Transformation rule

e_set("ip",long2ip(v("long"),default="xxx"))

Result

long: 4294967296 ip: xxx

4.5.8.7.12. Encoding and decoding functions

This topic describes the syntax and parameters of encoding and decoding functions. This topic also provides examples on how to use the functions. **Functions**

Category	Subcategory	Function	Description
	String	str_encode	Encodes a string.
		str_decode	Decodes a string.
	Dece 64	base64_encoding	Encodes data by using the Base64 algorithm.
	Base64	base64_decoding	Decodes data by using the Base64 algorithm.
For and in a section		html_encoding	Encodes data in the HTML format.
Encoding and decoding	HIML	html_decoding	Decodes HTML-encoded data.
		url_encoding	Encodes URL data.
	URL	url_decoding	Decodes URL data.
	JSON Web Token (JWT)	jwt_encoding	Encodes JSON data based on the JWT standard.
		jwt_decoding	Decodes data to raw JSON data based on the JWT standard.
	Gzip	gzip_compress	Compresses and encodes data.
Compression and		gzip_decompress	Decompresses compressed data.
decompression	Zlib	zlib_compress	Compresses and encodes data.
		zlib_decompress	Decompresses compressed data.
Encountion and docruption	Advanced Encryption Standard	aes_encrypt	Encrypts data by using the AES algorithm.
Encryption and decryption	(AES)	aes_decrypt	Decrypts data by using the AES algorithm.
	MD5	md5_encoding	Encodes data by using the MD5 algorithm.
Hash (digest)	SHA1	shal_encoding	Encodes data by using the SHA1 algorithm.
-	Cyclic redundancy check (CRC)	crc32_encoding	Calculates a CRC code for data.

str_encode

The str_encode function encodes a string by using a specified encoding format.

```
• Syntax
```

str_encode(value, "utf8", errors="ignore"))

• Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The value that you want to encode.
encoding	String	No	The encoding format. Default value: utf8. ASCII is supported.
errors	String	No	The method that is used to process characters if some characters cannot be recognized based on the encoding format. Valid values: • ignore (default): No characters are encoded. • strict: reports an error and discards the log that contains the characters. • replace: replaces the unrecognized characters with question marks (?). • xmlcharrefreplace: replaces the unrecognized characters with XML characters.

Response

An encoded string is returned.

• Examples

- Example 1
 - Raw log
 test: asewds
 - Transformation rule

e_set("f1", str_decode(str_encode("hello", "utf8"), "utf8"))

Result

test: asewds fl: hello

- Example 2
- Raw log

f2: test test data

Transformation rule

e_set("f1", str_encode(v("f2"), "ascii", errors="ignore"))

Result

f1:test f2:test test data

• Example 3

Raw log

f2: test test data

Transformation rule

e_set("f1", str_encode(v("f2"), "ascii", errors="strict"))

Result

An error is reported during execution.

- Example 4
 - Raw log

f2: test test data

Transformation rule

e_set("f1", str_encode(v("f2"), "ascii", errors="replace"))

Result

f1:test ???? f2:test test data

- Example 5
- Raw log

f2: test test data

Transformation rule

e_set("f1", str_encode(v("f2"), "ascii", errors="xmlcharrefreplace"))

Result

f1:test 测试数据 f2:test test data

str_decode

The str_decode function decodes an input value by using a specified encoding format.

• Syntax

str_decode(value, "utf8", errors="ignore"))

⑦ Note The str_decode function can process only the data of the byte data type.

Parameters

Parameter	Туре	Required	Description
value	Arbitrary (automatically converted to the string type)	Yes	The value that you want to decode.
encoding	Arbitrary (automatically converted to the string type)	No	The encoding format. Default value: utf8. ASCII is supported.

errors Arbitrary (automatically converted to the string type)	No	 The method that is used to process characters if some characters cannot be recognized based on the encoding format. Valid values: gnore (default): No characters are decoded. strict: reports an error and discards the log that contains the characters. replace: replaces the unrecognized characters with question marks (?). xmlcharrefreplace: replaces the unrecognized characters with XML characters.
--	----	---

Response

- A decoded value is returned.
- Example

0	Raw log	9	
	test:	asewds	

• Transformation rule

e_set("encoding", str_decode(b'\xe4\xbd\xa0\xe5\xa5\xbd', "utf8", 'strict'))

- Result
 - test: asewds encoding: hello

base64_encoding

The base64_encoding function encodes data by using the Base64 algorithm.

• Syntax

base64_encoding(value, format=None)

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value that you want to encode.
format	String	No	The Base64 encoding scheme. Valid values: RFC3548 and RFC4648 . Default value: RFC3548.

- Response
- An encoded string is returned.
- Example
 - Raw log

str_en : data to be encoded

• Transformation rule

e_set("str_base64",base64_encoding(v("str_en"))))

• Result

str_en : data to be encoded
str_base64 : ZGF0YSB0byBiZSB1bmNvZGVk

base64_decoding

The base64_decoding function decodes data by using the Base64 algorithm.

• Syntax

base64_decoding(value, format=None)

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value that you want to decode.
format	String	No	The Base64 decoding scheme. Valid values: RFC3548 and RFC4648 Default value: RFC3548. ? Note The Base64 decoding scheme in RFC 4648 uses equal signs (=) to pad a decoded value to a multiple of 4 bytes.

- Response
- A decoded string is returned.
- Example
- Raw log

str_de: ZGF0YSB0byBiZSB1bmNvZGVk

Transformation rule

e_set("str_de_base64",base64_decoding(v("str_de"))))

Result

str_de: ZGF0YSB0byBiZSB1bmNvZGVk
str_de_base64: data to be encoded

html_encoding

The html_encoding function encodes data in the HTML format.

Syntax

html_encoding(value)

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value that you want to encode.

Response

An encoded string is returned.

- Example
- Raw log

str :

• Transformation rule

e_set("str_html_en", html_encoding(v("str")))

• Result

str :

str_html_en :

html_decoding

The html_decoding function decodes HTML-encoded data.

Syntax

html_decoding(value)

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value that you want to decode.

Response

A decoded string is returned.

- Example
- Raw log

str :

• Transformation rule

e_set("str_html_de", html_decoding(v("str")))

Result

str :

str_html_de :

url_encoding

The url_encoding function encodes URL data.

Syntax

url_encoding(value)

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value that you want to encode.

Cloud Defined Storage

Response

An encoded string is returned.

- Example
- Raw log

content : https://www.example.org/hello/asdah

• Transformation rule

e_set("url",url_encoding(v("content")))

• Result

content : https://www.example.org/hello/asdah
url: https%3A%2F%www.example.org%2FHello%2Fasdah

url_decoding

The url_decoding function decodes URL data.

• Syntax

url_decoding(value)

Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value that you want to decode.

- Response
- A decoded string is returned.
- Example
 - Raw log

content : https%3A%2F%www.example.org%2FHello%2Fasdah

Transformation rule

e_set("URL",url_decoding(v("content"))))

Result

content : https%3A%2F%www.example.org%2FHello%2Fasdah URL : https://www.example.org/hello/asdah

jwt_encoding

The jwt_encoding function encodes JSON data based on the JWT standard.

ONOTE JWT is an open standard that defines a compact and self-contained method to transmit information between parties as JSON objects in a secure manner. For more information, see RFC 7519. The information is digitally signed and can be verified and trusted. JWTs can be signed by using a secret and the HMAC algorithm. JWTs can also be signed by using a public/private key pair and the RSA or ECDSA algorithm. For more information, see Introduction to JSON Web Tokens.

• Syntax

jwt_encoding(payload, key, algorithm="HS256", headers=None)

• Parameters

Parameter	Туре	Required	Description
payload	Dict	Yes	The value that you want to encode. The value must be in the JSON format. The JWT standard defines the following fields. You can also define custom fields. • iss: issuer • exp: expiration time • sub: subject • aud: audience • nbf: Not Before • iat: Issued At • jti: JWT ID The following sample code provides an example of raw JSON data: { "iss": "localhost", "sub": "name", "aud": "user", "aud": "user", "atreet": "street number", "city": "hangzhou", "country": "china" } }

key	String	Yes	The key that you want to use to encrypt a JWT. The key varies based on the encryption algorithm that you select. • Asymmetric encryption algorithm: PEM-encoded private key • Symmetric encryption algorithm: raw key
algorithm	String	No	The algorithm that you want to use to sign a JWT. Default value: HS256. Valid values: HS256, HS384, HS512, ES256, ES256K, ES384, ES512, RS256, RS384, RS512, PS256, PS384, PS512, and EdDSA .
headers	Dict	No	The information about the JWT header. Default value: { "typ": "JWT", "alg": "HS256" }

- Response
- An encoded string is returned.

Examples

- Example 1: Encode the value of the data field.
- Raw log

data: {"some": "payload"}

Transformation rule

e_set("jwt_token", jwt_encoding(v("data"),"secret", algorithm="HS256"))

Result

data:{"some": "payload"}

 $] \texttt{wt_token:eyJ0eXAiOiJKV1QiLCJhbGciOiJIUz11NiJ9.eyJzb21l1joicGF5bG9hZCJ9.Joh1R2dYzkRvDkqv3sygm5YyK8Gi4ShZqbhK2qxcs2U} a structure and stru$

• Example 2: Use an asymmetric encryption algorithm to encode the value of the data field.

Raw log

data: {"some": "payload"}

Transformation rule

```
e_set(
    "jwt_token",
    jwt_encoding(
        v("data"),
```

"----BEGIN RSA PRIVATE KEY-----

\nMIICYQIBAAKBgQCliaUr5cgShCn0127+w14XN297q/IviaewIeIJsKT2FlhBPsLn\nNIPsnqtQ9DFbjIyqyZvdmQFDJCSLpXaVc648yepnFDKbOfs3r+K4Crnpo2SuZmNV\nNDVE XlBz810zJYlwqVArM7qGAyCcRLBprwXB6wfEhk3CAP3c29+pwIDAQAB\nAoGAARo6519arbIbxx7fz7BEDAQMK0YaDGvbltg91S07cw4PPSYybNEG1BMKm01A\nV3v9BrR+u9PIDC5 wiODqEoSyk80wo1E2kWA6+MNclYfYjaJeiRJ5PzCud/\niUObonptRzxuTng+u1cGuX7QwUhwGJdXVBUAtJFYwXR2qVECRQC5S+6vdFESRL5X\n7yBZVM6+4912cdehMvOHwT17Us jeSbiogvv02HbYi1rW9ZydKsiAWP5vU\nS3L34CS811VcDQ19APr0117clxg5fX8wAWv2d+e+MfZoB3ohb8671W3pmp3JVnjY\ntzhoYNNQmmmQQWf7n3J63MQz4sYYNn0gwJEE Dw1MFnn0H/AOKt79LtKKG1\n+BFhSqbBFDzWmvBu4Fo9oQ3Lr63gzSCGSrb6JhkCIptz5hIJmOARozwdeebVozkC\nPQDwOqVmU3c/P8nB66RiGditw1Jt1yTaSW6jkOyUc73iRngq HGd1kwwDX4/\nWfoAyEha1Y6Fh5s1CosCRQCy4b3hnQws5zz/gmGNnoyxn9N+A09ySaFtn2WkU1rR\nZ6DwoJz+n6EgjLY8z6ZQyv342iob05ZKkZHFv09QYGqk7y81JA==\n----END R5A PRIVATE KEY-----\n",

algorithm="RS256",

Result

),

data:{"some": "payload"}

jwt_token:eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzb2111joicGF5bG9hZCJ9.Ewwls5YPuJCmAR3XR2tcptOLrH83wVCzmUaUpGMzMLcPknRrIvbDmFGlN1Qha-PM x0jsxt3t1oxpz7P3z3SR9o4qyWusAb99UG_Jn8oP8W0a5GKSy4UEJB0xgpVvJ15F2JaIPeUSHpV0VeS2WAsGSBBSAa0Mkrc-8uie-H4J9M0

Cloud Defined Storage

- Example 3: Set the algorithm parameter to HS256 and configure the headers parameter to encode the value of the data field.
- Raw log

.

data: {"some": "payload"}

Transformation rule
e_set(
"jwt_token",
jwt_encoding(
v("data"),
"secret",
algorithm="HS256",
headers={"kid": "230498151c214b788dd97f22b85410a5"},
),
)

Result

data:{"some": "payload"}

jwt_token:eyJUeXiOiJKVlQiLCJhbGciOiJIUzI1NiIsImtpZCI6IjIzMDQ5ODE1MWMyMTRiNzg4ZGQ5N2YyMmI4NTQxMGE1In0.eyJzb21lIjoicGF5bG9hZCJ9.gdQ884yj1nL fQaClE6rJC2x8v2OP2s_eXOLhZA

• Example 4: Specify a custom JWT field for the data field and encode the value of the data field.

Raw log

data: {"some": "payload", "iss": 9, "sub": "name", "nbf": 123, "iat": "22"}

Transformation rule

e_set("jwt_token", jwt_encoding(v("data"),"secret"))

Result

data:{"some": "payload", "iss": 9, "sub": "name", "nbf": 123, "iat": "22"}
jwt_token:eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzb21lIjoicGF5bG9hZCIsIm1zcyI6OSwic3ViIjoibmFtZSIsIm5iZiI6MTIzLCJpYXQiOiIyMiJ9.c6MxZHdXsg
1Mdqj208NO_3rNGjYnvo6c7HNAyk

jwt_decoding

The jwt_decoding function decodes data to raw JSON data based on the JWT standard.

ONOTE JWT is an open standard that defines a compact and self-contained method to transmit information between parties as JSON objects in a secure manner. For more information, see RFC 7519. The information is digitally signed and can be verified and trusted. JWTs can be signed by using a secret and the HMAC algorithm. JWTs can also be signed by using a public/private key pair and the RSA or ECDSA algorithm. For more information, see Introduction to JSON Web Tokens.

• Syntax

jwt_decoding(jwt_payload, key="", algorithms=None, options=None, audience=None, issuer=None, leeway=0)

Parameters

Parameter	Туре	Required	Description
jwt_payload	String	Yes	The value that you want to decode.
key	String	Yes	The key that you want to use to decrypt data. This key must be the same as the key that is used for encoding.
algorithm	List	No	The algorithm that you want to use to sign a JWT. Example: HS256.
options	Dict	No	The extended options for decoding and verification. Valid values: • verify_signature: specifies whether to verify the encrypted JWT signature. Default value: True. • require: specifies the fields that must exist in the encoded data. By default, this option is empty. Example: ["exp", "iat", "hbf"]. • verify_aud: specifies whether to check the aud field. Default value: True. • verify_iss: specifies whether to check the iss field. Default value: True. • verify_exp: specifies whether to check the exp field. Default value: True. • verify_at: specifies whether to check the iat field. Default value: True. • verify_iat: specifies whether to check the hof field. Default value: True. • verify_nbf: specifies whether to check the nbf field. Default value: True. • verify_nbf: specifies whether to check the nbf field. Default value: True. Example: ["require": ["aud"], "verify_aud": True]
audience	String/List	No	Checks the value of the aud field. You must configure this parameter if the aud field is specified for encoding and "require": ["aud"] or "verify_aud": True is specified for the options parameter.

issuer	String	No	Checks the value of the iss field. You must configure this parameter if the iss field is specified for encoding and "require": ["iss"] or "verify_iss": True options parameter.
leeway	float	No	The interval at which an expiration check is performed. Unit: seconds.

Response

A decoded string is returned.

• Examples

• Example 1: Set the signature algorithm for decryption to HS256 to decode the value of the data field.

Raw log

data:eyJ0eXAi0iJKV1QiLCJhbGci0iJIUzI1NiJ9.eyJzb21lIjoicGF5bG9hZCJ9.Joh1R2dYzkRvDkqv3sygm5YyK8Gi4ShZqbhK2gxcs2U

Transformation rule

e_set("data_decoded", jwt_decoding(v("data"), "secret", algorithms="HS256"))

Result

data_decoded: {"some": "payload"}

• Example 2: Use a public key to decode the value of the data field.

Raw log

data: "eyJ0eXAiOiJKV1qiLCJhbGciOiJSUzI1NiJ9.eyJzb21l1joicGF5bG9h2CJ9.Ewwls5YPuJCmAR3XR2tcpt0LrH83wVCzmUaUpGMzMLcPknRrIvbDmFG1N1qha-cpt0LrH83wVCzmUaUpGMzMLcPknRrIvbNA40mA40a-cpt0LrH83wVCzmUaUpGMzMLcPknRrIvbNA40a-cpt0LrH83wVCzmUaUpGMzMLcPknRrIvbNA40a-cpt0LrH83wVCzmUaUpGMzMA40a-cpt0LrH83wVCzmUaUpGMzMA40a-cpt0LrH83wVCzmUaUpGMzMA40a-cpt0LrH83wVCzmUaUpGMzMLcPknRrIvbNA40a $\texttt{PMx0jsxt3tloxpz7P3z3SR9o4qyWusAb99UG_Jn8oP8W0a5GKSy4UEJB0xgpVvJ15F2JaIPeUSHpV0VeS2WAsGSBBSAaOMkrc-8uie-H4J9M0"}$

Transformation rule

```
e_set(
   "data_decoded",
   jwt_decoding(
       v("data"),
        "----BEGIN RSA PUBLIC KEY-----
\nMIGJAoGBALWJpSvlyBKEKfTXbv7DXhc3b3ur8i+Jp7Ah4gmwpNkXWEE+wuc0g+ye\nq1D0MVuMjKrJm92ZAUMkJIuldpVzrjzJ6mcUMps5+zev4rgKuemjZK5mY1U0NUSL\nikQJ
XTMljXCpUCszuoYDIJxEsGmvBcHrB8SGTcIA/dzb36nAgMBAAE=\n----END RSA PUBLIC KEY-----\n",
```

```
algorithms="RS256",
```

```
),
)
```

Result

```
data_decoded: {"some": "payload"}
```

data:"eyJ0eXAiOiJKVlQiLCJhbGciOiJSUzIlNiJ9.eyJzb21lIjoicGF5bG9hZCJ9.Ewwls5YPuJCmAR3XR2tcpt0LrH83wVCzmUaUpGMzMLcPknRrIvbDmFGlNlQha-PMx0jsxt3tloxpz7P3z3SR9o4qyWusAb99UG_Jn8oP8W0a5GKSy4UEJB0xgpVvJ15F2JaIPeUSHpV0VeS2WAsGSBBSAaOMkrc-8uie-H4J9M0"

• Example 3: Use the default value of the options parameter to decode the value of the data field.

Raw log

data:

"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzb211IjoicGF5bG9hZCIsImlzcyI60Swic3ViIjoibmFtZSIsIm5iZiI6MTIzLCJpYXQiOiIyMiJ9.DzvqhJd0PrTFk6eeASG trLBt H3xC7CqOATRRw"

Transformation rule

```
e_set(
    "data_decoded", jwt_decoding(v("data"), "secret", algorithms="HS256", options=None)
```

)

Result

```
data_decoded:{
        "some": "payload",
        "iss": 9,
        "sub": "name",
        "nbf": 123,
        "iat": "22"
```

data: "eyJ0eXAi0iJKV1QiLCJhbGci0iJIUz11NiJ9.eyJzb21l1joicGF5bG9h2CIsIm1zcyI60Swic3ViIjoibmFtzSIsIm5iZiI6MT1zLCJpYXQi0iIyMiJ9.DzvqhJd0PrTFk6ologianterseteen and the set of theZxOoDtrLBt H3xC7CqOATRRw

Cloud Defined Storage

• Example 4: Specify a custom value for the options parameter to decode the value of the data field.

```
    Raw log
```

data: "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzb21lIjoicGF5bG9hZCIsImlzcyI6Im5hbWUifQ.XwT9jqwofcdSP6olidbiYPC6CnZd36OEqCHZmGmooWM"

•	Transformation rule
	e_set(
	"data_decoded",
	jwt_decoding(
	v("data"),
	"secret",
	algorithms="HS256",
	<pre>options={"require": ["iss"], "verify_iss": True},</pre>
	issuer="name",
),
)

Result

data_decoded:{"some": "payload", "iss": "name"}

data: "eyJ0eXAiOiJKVlQiLCJhbGciOiJIUzIlNiJ9.eyJzb2llIjoicGF5bG9hZCIsImlzcyI6Im5hbWUifQ.XwT9jqwofcdSP6olidbiYPC6CnZd360EqCHZmGmooMM"

gzip_compress

The gzip_compress function compresses and encodes data.

Syntax

gzip_compress(data, compresslevel=6, to_format="base64", encoding="utf-8")

Parameters

Parameter	Туре	Required	Description
data	String	Yes	The data that you want to compress.
compresslevel	Int	No	 The compression level. Valid values: 0 to 9. Default value: 6. 1: The compression speed is the highest, and the compression ratio is the lowest. 9: The compression speed is the lowest, and the compression ratio is the highest. 0: Data is not compressed.
to_format	String	No	The encoding format for the compressed data. Valid values: base64 and hex.
encoding	String	No	The encoding format for the raw data. Default value: utf-8. For more information about other encoding formats, see Standard encoding formats.

Response

An encoded string is returned.

Examples

 $\circ~$ Example 1: Compress a log field and encode the field by using the Base64 algorithm.

Raw log

content: I always look forward to my holidays whether I travel or stay at home.

Transformation rule

e_set("base64_encode_gzip_compress",gzip_compress(v("content"),to_format="base64"))

```
    Result
```

content: I always look forward to my holidays whether I travel or stay at home. base64_encode_gzip_compress:

H4SIAA8JX14C/xXK0QmAMAwFwFXeB07RMQKNREx5kAZDtle/7wbES3rDyRsnoyQmklgNo1/ztzJN08BAhjzqYGCnNCS/tPR4AcgrnWVGAAAA

$\circ~$ Example 2: Encode a log field by using the HEX encoding format.

Raw log

content: H4sIAA8JXl4C/xXK0QmAMAwFwFXeB07RMQKNREx5kAZDtle/7wbES3rDyRsnoyQmklgNo1/ztzJN08BAhjzqYGCnNCS/tPR4AcgrnWVGAAAA

Transformation rule

e_set("hex_encode_gzip_compress", gzip_compress(v("content"), to_format="hex"))

Result

content:H4sIAA8JX14C/xXK0QmAMAwFwFXeB07RMQKNREx5kAZDtle/7wbES3rDyRsnoyQmklgNo1/ztzJN08BAhjzqYGCnNCS/tPR4AcgrnWVGAAAA
hex_encode_gzip_compress:1f8b08004a478c6202ff0dc1dd0e43301800d047aa65156e3ff52f4a2ba17649c43255194d4a9f9e73527c64007e2e2426e81485c35628c1c
535079bc6405e5d1e92ef009b59c906786a879efe1c50fb55d6c5de44cb717b2dae6d4f103f8feecbf4f88a2a441bae618c679575d9bc0e306907876806c000000

gzip_decompress

The gzip_decompress function decompresses compressed data.

• Syntax

gzip_decompress(data, from_format="base64", encoding="utf-8")

Parameters

Parameter	Туре	Required	Description
data	Arbitrary	Yes	The data that you want to decompress.
from_format	String	No	The encoding format for the compressed data. Valid values: base64 and hex.
encoding	String	No	The encoding format for the raw data. Default value: utf-8. For more information about other encoding formats, see Standard encoding formats.

Response

Decompressed data is returned.

Examples

• Example 1: Decode a log field by using the Base64 algorithm.

Raw log

content: H4sIAA8JX14C/xXK0QmAMAwFwFXeB07RMQKNREx5kAZDtle/7wbES3rDyRsnoyQmklgNo1/ztzJN08BAhjzqYGCnNCS/tPR4AcgrnWVGAAAA

Transformation rule

e_set("gzip_decompress", gzip_decompress(v("content"), from_format="base64"))

Result

content: H4sIAA8JX14C/xXK0QmAMAwFwFXeB07RMQKNREx5kAZDtle/7wbES3rDyRsnoyQmklgNo1/ztzJN08BAhjzqYGCnNCS/tPR4AcgrnWVGAAAA gzip_decompress: I always look forward to my holidays whether I travel or stay at home.

$\circ~$ Example 2: Decode a log field by using the HEX decoding format.

Raw log

content:1f8b0800bff8856202ff0dc1dd0e43301800d047aa65156e3ff52f4a2ba17649c43255194d4a9f9e73527c64007e2e2426e81485c35628c1c42616535079bc6405992ef009b59c906786a879efe1c50fb55d6c5de44cb717b2dae6d4f103f8feecbf4f88a2a441bae618c679575d9bc0e306907876806c000000

Transformation rule

e_set("gzip_decompress", gzip_decompress(v("content"), from_format="hex"))

Result

content:1f8b0800bff8856202ff0dc1dd0e43301800d047aa65156e3ff52f4a2ba17649c43255194d4a9f9e73527c64007e2e2426e81485c35628c1c42616535079bc6405 92ef009b59c906786a879efe1c50fb55d6c5de44cb717b2dae6d4f103f8feecbf4f88a2a441bae618c679575d9bc0e306907876806c000000 gzip_decompress:H4sIAA8JX14C/xXK0QmAMAwFwFXeB07RMQKNREx5kAZDtle/7wbES3rDyRsnoyQmklgNo1/ztzJN08BAhjzqYGCnNCS/tPR4AcgrnWVGAAAA

zlib_compress

The zlib_compress function compresses and encodes data.

Syntax

zlib_compress(data, compresslevel=6, to_format="base64", encoding="utf-8")

• Parameters

Parameter	Туре	Required	Description
data	String	Yes	The data that you want to compress.
compresslevel	Int	No	 The compression level. Valid values: 0 to 9. Default value: 6. 1: The compression speed is the highest, and the compression ratio is the lowest. 9: The compression speed is the lowest, and the compression ratio is the highest. 0: Data is not compressed.
to_format	String	No	The encoding format for the compressed data. Set the value to base64.
encoding	String	No	The encoding format for the raw data. Default value: utf-8. For more information about other encoding formats, see Standard encoding formats.

Response

Encoded data is returned.

• Example

Raw log

content: I always look forward to my holidays whether I travel or stay at home.

• Transformation rule

e_set("zlib_compress", zlib_compress(v("content"), to_format="base64"))

• Result

zlib_compress: "eJwVytEJgDAMBcBV3gTu0TECjURMeZAGQ7ZXv+8GxEt6w8kbJ6MkJpJYDANf87cyTdPAQIY86mBgpzQkv7T0eAGNshln"
content: "I always look forward to my holidays whether I travel or stay at home."

zlib_decompress

The zlib_decompress function decompresses compressed data.

Syntax

zlib_decompress(data, from_format="base64", encoding="utf-8")

• Parameters

Parameter	Туре	Required	Description
data	String	Yes	The data that you want to decompress.
from_format	String	No	The encoding format for the compressed data. Set the value to base64.
encoding	String	No	The encoding format. Default value: utf-8. For more information about other encoding formats, see Standard encoding formats.

Response

Decompressed data is returned.

- Example
- Raw log

content: "eJwVytEJgDAMBcBV3gTu0TECjURMeZAGQ7ZXv+8GxEt6w8kbJ6MkJpJYDANf87cyTdPAQIY86mBgpzQkv7T0eAGNshln"

• Transformation rule

e_set("zlib_decompress", zlib_decompress(v("content"), from_format="base64"))

• Result

content: "eJwVytEJgDAMBcBV3gTu0TECjURMeZAGQ7ZXv+8GxEt6w8kbJ6MkJpJYDANf87cyTdPAQIY86mBgpzQkv7T0eAGNshln"
zlib_decompress: "I always look forward to my holidays whether I travel or stay at home."

aes_encrypt

The aes_encrypt function encrypts data by using the AES algorithm. The AES algorithm is a common symmetric encryption algorithm that is used. To improve data security, you can use the AES algorithm to encrypt data.

Syntax

aes_encrypt(data, key, mode, pad_style, pad_block, input_format, input_encoding, output_format, iv)

• Parameters

Parameter	Туре	Required	Description
data	String	Yes	The data that you want to encrypt.
key	String	Yes	The key that you want to use to encrypt data.
mode	String	No	The AES encryption mode. • CBC (default): Cipher Block Chaining • ECB: Electronic Codebook Book • CFB: Cipher Feedback • OFB: Output Feedback • CTR: Counter • OPENPGP
pad_style	String	No	The padding mode. Default value: pkcs7. Valid values: iso7816, x923, and pkcs7.
input_format	String	No	The format of input characters. Default value: raw. Valid values: • raw: bytes • hex: hexadecimal • base64: Base64 encoded
input_encoding	String	No	The encoding format for input characters. This parameter is required only if you set the input_format parameter to raw. Default value: uft-8.

output_format	String	No	The format of output characters. Default value: hex. Valid values: • raw: bytes • hex: hexadecimal • base64: Base64 encoded
iv	Bytes	No	The offset that you want to use for encryption.

Response

An encrypted string is returned.

- Examples
- Example 1
- Raw log
 - "test": "aliyuntest"
- Transformation rule

e_set('result',aes_encrypt(v("test"), "gwertyuiopasdfgd", iv=b"xxywosjdapdiawdk", output_format="base64"))

Result

"result": "gXIqu0cBBtZHQxJBK8GLeA=="

- Example 2
 - Raw log

"test": "aliyuntest"

Transformation rule

e_set('result',aes_encrypt(v("test"), "gwertyuiopasdfgh", iv=b"ywisnjaduaqibdqi", mode="OFB"))

Result

"result": "5cac3e9e1c42f713dc6d"

aes_decrypt

The aes_decrypt function decrypts data by using the AES algorithm.

Syntax

aes_decrypt(data, key, mode, pad_style, input_format, input_encoding, output_format, iv, output_encoding)

• Parameters

Parameter	Туре	Required	Description
data	String	Yes	The data that you want to decrypt.
key	String	Yes	The key that you want to use to decrypt data.
mode	String	No	The AES decryption mode. • CBC (default): Cipher Block Chaining • ECB: Electronic Codebook Book • CFB: Cipher Feedback • OFB: Output Feedback • CTR: Counter • OPENPGP
pad_style	String	No	The padding mode. Default value: pkcs7. Valid values: iso7816, x923, and pkcs7.
input_format	String	No	The format of input characters. Default value: hex. Valid values: • raw: bytes • hex: hexadecimal • base64: Base64 encoded
input_encoding	String	No	The encoding format for input characters. This parameter is required only if you set the input_format parameter to raw. Default value: uft-8.
output_format	String	No	The format of output characters. Default value: raw. Valid values: • raw: bytes • hex: hexadecimal • base64: Base64 encoded
iv	Bytes	No	The offset that you want to use for decryption.
output_encoding	String	No	The encoding format for output characters. Default value: None.

Response

A decrypted string is returned.

Examples

- Example 1
 - Raw log

"test": "gXIqu0cBBtZHQxJBK8GLeA=="

Transformation rule

e_set('result', aes_decrypt(v("test"), "qwertyuiopasdfgd", iv=b"xxywosjdapdiawdk", input_format="base64"))

Result

"result": "aliyuntest"

- Example 2
- Raw log

"test": "5cac3e9e1c42f713dc6d"

Transformation rule

e_set('result', aes_decrypt(v("test"), "qwertyuiopasdfgh", iv=b"ywisnjaduaqibdqi", mode="OFB"))

Result

"result": "aliyuntest"

md5_encoding

The md5_encoding function encodes data by using the MD5 algorithm.

• Syntax

md5_encoding(value, format="hex")

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value that you want to encode.
format	String	No	Valid values: binary and hex. Default value: hex.

Response

An encoded string is returned.

- Examples
 - Example 1
 - Raw log
 - str : GeeksforGeeks
 - Transformation rule

e_set("str_md5_en",md5_encoding(v("str")))

Result

```
str : GeeksforGeeks
str_md5_en : fle069787ece74531d112559945c6871
```

- Example 2
 - Raw log

str : GeeksforGeeks

Transformation rule

e_set("str_md5_en",base64_encoding(md5_encoding(v("str"), format="binary")))

Result

str : GeeksforGeeks
str_md5_en : 8eBpeH70dFMdESVZlFxocQ==

sha1_encoding

The shal_encoding function encodes data by using the SHA1 algorithm.

• Syntax

shal_encoding(value, format=None)

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The value that you want to encode.
format	String	No	The encoding type. Valid values: SHA1, SHA224, SHA256, SHA384, and SHA512. Default value: SHA1.

Response

An encoded string is returned.

• Example
Raw log

str : GeeksforGeeks

• Transformation rule

e_set("str_shal", shal_encoding(v("str"))) e_set("str_sha512", sha1_encoding(v("str"), format='SHA512'))
e_set("str_sha224", sha1_encoding(v("str"), format='SHA224')) e_set("str_sha384", sha1_encoding(v("str"), format='SHA384')) e_set("str_sha256", sha1_encoding(v("str"), format='SHA256'))

• Result

str : GeeksforGeeks str_sha1 : 4175a37afd561152fb60c305d4fa6026b7e79856 str sha512 : - 0.4 0.4 553500df254a75db63d1baa35ad99f26f1b399c31f3c666a7fc67ecef3bdcdb7d60e8ada90b722 str_sha224 : 173994f309f727ca939bb185086cd7b36e66141c9e52ba0bdcfd145d

str_sha384 : d1e67b8819b009ec7929933b6fc1928dd64b5df31bcde6381b9d3f90488d253240490460c0a5a1a873da8236c12ef9b3

str sha256 : f6071725e7ddeb434fb6b32b8ec4a2b14dd7db0d785347b2fb48f9975126178f

crc32_encoding

The crc32_encoding function calculates a CRC code for data.

• Syntax

crc32_encoding(data, input_format="raw", input_encoding="utf-8")

• Parameters

Parameter	Туре	Required	Description
data	String	Yes	The value for which you want to calculate a CRC code.
input_format	String	No	The format of input characters. Default value: raw. Valid values: • raw: bytes • hex: hexadecimal • base64: Base64 encoded
input_encoding	String	No	The encoding format for input characters. This parameter is required only if you set the input_format parameter to raw. Default value: uft-8.

- Response
- The CRC code for the input value is returned.
- Examples

• Example 1: Calculate a CRC code for the test field.

- Raw log
 - test: aliyuntest
- Transformation rule

e_set("str_crc32", crc32_encoding(v("test")))

Result

str_crc32:1434103726 test:aliyuntest

• Example 2: Join the test1 field and test2 field, encode the fields by using the MD5 algorithm, and calculate a CRC code for the fields.

- Raw log
 - test1: test1 test2: test2
- Transformation rule

```
e_set(
```

```
"str crc32",
crc32 encoding(
```

md5_encoding(str_join("+", v("test1"), v("test2")), format="binary")),

```
)
```

Result

```
str_crc32:369733261
test1:test1
test2:test2
```

- Example 3: Calculate a CRC code for the test field. The field value is Base64-encoded.
- Raw log

test: Taloz+e+PzP3NltrEXiCig==

Transformation rule

e_set("str_crc32", crc32_encoding(v("test"), input_format="base64"))

Result

str_crc32:1093789404
test:Taloz+e+PzP3NltrEXiCig==

4.5.8.7.13. List functions

This topic describes the syntax and parameters of list functions. This topic also provides examples on how to use the functions.

Functions

Function	Description
lst_make	Creates a list.
lst_insert	Inserts elements to a specified position in a list.
lst_append	Appends elements to a list.
lst_delete_at	Deletes the element at a specified position from a list.
lst_reverse	Reverses the order of elements in a list.
op_slice	Returns specific elements.
lst_get	Returns the element at a specified position in a list or a tuple.
op_len	Calculates the number of elements in a list or a tuple.

lst_make

The lst_make function creates a list.

Syntax

lst_make(value1, value2, ...)

• Parameters

Parameter	Туре	Required	Description
value1	String	Yes	The element of the list.
value2	String	Yes	The element of the list.

- Response
- The created list is returned.
- Example
- Raw log
 - content:test
- Transformation rule

e_set("hello", lst_make("k1","k2"))

• Result

content:test
hello:["k1", "k2"]

lst_insert

The lst_insert function inserts elements to a specified position in a list.

Syntax

lst_insert(list_string, location, value1, value2, ...)

• Parameters

Parameter	Туре	Required	Description
list_string	List	Yes	The input list.
location	Number	Yes	The position to which you want to insert elements.
valuel	String	Yes	The element that you want to insert.
value2	String	No	The element that you want to insert.

Response

The list to which the elements are inserted is returned.

- Example
 - Raw log
- ctx: ["k1","k2"] • Transformation rule
 - e_set("hello", lst_insert(v("ctx"), 0, "k0"))
- Result

ctx: ["k1","k2"] hello: ["k0", "k1", "k2"]

lst_append

The lst_append function appends elements to a list.

• Syntax

lst_append(list_string, value1, value2, ...)

Parameters

Parameter	Туре	Required	Description
list_string	List	Yes	The input list.
value1	String	Yes	The element that you want to append.
value2	String	No	The element that you want to append.

Response

The list to which the elements are appended is returned.

- Example
- Raw log

ctx: ["k1","k2"]

Transformation rule

e_set("hello", lst_append(v("ctx"), "k3"))

• Result

ctx: ["k1","k2"] hello: ["k1", "k2", "k3"]

lst_delete_at

The lst_delete_at function deletes the element at a specified position from a list.

• Syntax

lst_delete_at(list_string, location)

Parameters

Parameter	Туре	Required	Description
list_string	list	Yes	The input list.
location	Number	Yes	The position of the element that you want to delete. The position of the first element is 0.

Response

The list from which the element is deleted is returned.

• Example

Raw log

- ctx: ["k1","k2"]
- Transformation rule
 - e_set("hello", lst_delete_at(v("ctx"),1))
- Result

ctx: ["k1","k2"] hello: ["k1"]

lst_reverse

The lst_reverse function reverses the order of elements in a list.

Syntax

Parameter	Туре	Required	Description	
Parameters				
<pre>lst_reverse(list_string)</pre>				
•				

4

Cloud Defined Storage

User Guide-Log Service

	list_string	List		Yes	The input list.		
• R	lesponse						
Т	he list whose order of eleme	nts is reversed is retu	irned.				
• E	• Example						
0	• Raw log						
	ctx: ["v1","v2"]						
0	Transformation rule						
	e_set("hello", lst_reverse	e(v("ctx")))					
٥	Result						
	ctx: ["v1","v2"] hello: ["v2","v1"]						
lst	aet						
The	e lst get function returns the	element at a specifie	d positio	n in a list or a tuple.			
• 5	yntax						
	<pre>lst_get(list_string, location)</pre>	on)					
• P	arameters						
	Parameter	Туре		Required	Description		
	list_string	List		Yes	The input list.		
	location	Int		Yes	The position of the element that you want to obtain. The position of the first element is 0. For example, if the input list is $["a","b","c"]$, you can obtain the elements at the following positions: 0, 1, and 2.		
• R	lesponse						
т	he element at the specified p	position is returned.					
• E	xample						
٥	Raw log						
	ctx: ["v1","v2"]						
0	Transformation rule						
	<pre>e_set("hello", lst_get(v()</pre>	"ctx"),1))					
٥	Result						
	ctx: ["v1","v2"] hello: "v2"						
4.	5.8.7.14. Dictio	nary functio	ons				
This	This topic describes the syntax and parameters of dictionary functions. This topic also provides examples on how to use the functions.						
Fu	Inctions						
F	unction	c.	Descriptio	on			

Function	Description
dct_make	Constructs a dictionary.
dct_update	Updates a dictionary.
dct_delete	Deletes key-value pairs from a dictionary.
dct_keys	Returns the keys of a dictionary.
dct_values	Returns the values of a dictionary.
dct_get	Returns the value that corresponds to a specified key in a dictionary.
op_len	Returns the number of elements in a dictionary.

dct_make

The dct_make function constructs a dictionary.

• Syntax

	dct_	<pre>dct_make(key1, value1, key2, value2,)</pre>					
	?	Note	You must speci	fy the key and value paramete	ers in pairs.		
•	Paran	neters					
	Para	ameter		Туре	Required	Description	

User Guide-Log Service

key	String	Yes	The key in the dictionary that you want to construct.
value	String	Yes	The value in the dictionary that you want to construct.

- Response
 - The constructed dictionary is returned.
- Example
 - Raw log content:test
- Transformation rule

e_set("hello", dct_make("k1","v1","k2","v2"))

• Result

content:test hello:{"k1": "v1", "k2": "v2"}

dct_update

The dct_update function updates a dictionary.

• Syntax

dct_update(dict1, dict2)

• Parameters

Parameter	Туре	Required	Description
dict1	dict	Yes	The dictionary that you want to update.
dict2	dict	Yes	The dictionary information that is added to the specified dictionary.

Response

The updated dictionary is returned.

- Example Raw log
- - ctx: {"k1":"v1","k2":"v2"}

• Transformation rule

e_set("hello", dct_update(v("ctx"), {"k3": "v3"}))

Result

ctx: {"k1":"v1","k2":"v2"} hello: {"k1": "v1", "k2": "v2", "k3": "v3"}

dct_delete

The dct_delete function deletes key-value pairs from a dictionary.

Syntax

dct_delete(dict, key1, key2, ...)

• Parameters

Parameter	Туре	Required	Description
dict	dict	Yes	The dictionary from which you want to delete the specified key- value pair.
keyl	String	Yes	The key of the key-value pair that you want to delete from the dictionary.
key2	String	No	The key of the key-value pair that you want to delete from the dictionary.

Response

The dictionary from which the specified key-value pairs are deleted is returned.

- Example
- Raw log

ctx: {"k1":"v1","k2":"v2"}

• Transformation rule

e_set("hello", dct_delete(v("ctx"), "k2"))

Result

ctx: {"k1":"v1","k2":"v2"}
hello: {"k1":"v1"}

dct_keys

The dct_keys function returns the keys of a dictionary.

• Syntax

dct_keys(dict)

• Parameters

Parameter	Туре	Required	Description
dict	dict	Yes	The dictionary from which you want to obtain keys.

Response

The keys of the dictionary are returned.

• Example

Raw log

ctx: {"k1":"v1","k2":"v2"}

• Transformation rule

e_set("hello", dct_keys(v("ctx")))

Result

ctx: {"k1":"v1","k2":"v2"} hello: ["k1","k2"]

dct_values

The dct_values function returns the values of a dictionary.

Syntax

dct_values(dict)

Parameters

Parameter	Туре	Required	Description
dict	dict	Yes	The dictionary from which you want to obtain values.

Response

The values of the dictionary are returned.

- Example
- Raw log
 - ctx: {"k1":"v1","k2":"v2"}
- Transformation rule

e_set("hello", dct_values(v("ctx")))

• Result

```
ctx: {"k1":"v1","k2":"v2"}
hello: ["v1","v2"]
```

dct_get

The dct_get function returns the value that corresponds to a specified key in a dictionary.

• Syntax

dct_get(dict,key,default=None)

• Parameters

Parameter	Туре	Required	Description
dict	dict	Yes	The dictionary from which you want to obtain the value of the specified key.
key	String	Yes	The key whose value you want to obtain.
default	String	No	The value that is returned if the specified key does not exist.

Response

The value that corresponds to a specified key in a dictionary is returned.

• Examples

- Example 1
 - Raw log

ctx: {"k1":"v1","k2":"v2"}

Transformation rule

e_set("hello", dct_get(v("ctx"), "k1"))

Result

ctx: {"k1":"v1","k2":"v2"}
hello: v1

 $\circ~$ Example 2: The specified key does not exist. The value of the default parameter is returned.

Raw log

ctx: {"k1":"v1","k2":"v2"}

Transformation rule

e_set("hello", dct_get(v("ctx"), "k3",default="123"))

Result

ctx: {"k1":"v1","k2":"v2"} hello: 123

4.5.8.7.15. Table functions

This topic describes the syntax and parameters of table functions. This topic also provides examples on how to use the functions.

Functions

Category	Function	Description
Text to table	tab_parse_csv	Constructs a table from CSV-formatted text.
Table to dictionary	tab_to_dict	Constructs a dictionary from a table.

tab_parse_csv

The tab_parse_csv function constructs a table from CSV-formatted text.

Syntax

tab_parse_csv(data, sep=',', quote='"', lstrip=True, headers=None, case_insensitive=True)

Parameters

Parameter	Туре	Required	Description
data	String	Yes	The CSV-formatted text.
sep	String	No	The delimiter that is used in the CSV-formatted text. The default value is a comma (,).
quote	String	No	The quote. If a value contains the delimiter, you must use the quote to enclose the value. The default value is double quotation marks (").
lstrip	Bool	No	Specifies whether to remove the spaces from the left of each keyword. Default value: True.
headers	String\String List	No	The headings that are obtained after parsing. By default, the system retrieves the headings from the first line of the CSV- formatted text. If the first line does not store headings, you can configure this parameter to pass headings to the function. The value of this parameter can be a string or a string list.
case_insensitive	Bool	No	Specifies whether data is not case-sensitive when the system performs mapping. Default value: True.

Response

The table that you construct is returned.

• Examples

- $\circ~$ Example 1: Construct a table and map the value of a field to the table.
 - Raw log
 - city:nanjing

Transformation rule

"city", "province",

-) • Result

city:nanjing province:jiangsu

• Example 2: Construct a table and map the values of multiple fields to the table.

Raw log

city:nanjing province:jiangsu

Transformation rule

```
e_table_map(
   tab_parse_csv(
        "province,city,pop,gdp\nshanghai,shanghai,2000,1000\njiangsu,nanjing,800,500"
),
   ["province", "city"],
   ["pop", "gdp"],
```

) • Result

```
city:nanjing
gdp:500
pop:800
province:jiangsu
```

- Example 3: Construct a table and map the values of multiple fields to the table. The field names are different from the column names in the table. In the parentheses that include the source fields, the first field is a field of the raw log and the second field is a field of the table. In the parentheses that include the destination fields, the first field is a field of the table and the second field is a new field that is returned.
 - Raw log

```
city:nanjing
province:jiangsu
```

Transformation rule

```
e_table_map(
tab_parse_csv(
```

- "prov,city,pop,gdp\nshanghai,shanghai,2000,1000\njiangsu,nanjing,800,500"
-),
- [("province", "prov"), "city"], [("pop", "population"), ("gdp", "GDP")],
-)

Result

```
GDP:500
city:nanjing
population:800
province:jiangsu
```

tab_to_dict

The tab_to_dict function constructs a dictionary from a table.

• Syntax

tab_to_dict(table, key_field, value_field, key_join=",", value_join=",")

```
• Parameters
```

Parameter	Туре	Required	Description
table	Table	Yes	The data in the table.
key_field	String\String List	Yes	The columns that are used to construct keys in the dictionary. Connect multiple columns with the character specified by $\tt key_join$.
value_field	String\String List	Yes	The columns that are used to construct values in the dictionary. Connect multiple columns with the character specified by <code>value_join</code> .
key_join	String	No	The string that is used to connect multiple columns. The columns are used as keys in the dictionary. The default value is a comma (,).

User Guide-Log Service

Cloud Defined Storage

value_join	String	No	The string that is used to connect multiple columns. The columns are used as values in the dictionary. The default value is a comma (,).
Response The dictionary that you co Examples • Example 1 • Raw log	nstruct is returned.		
k1:v1 city:nj			
 Transformation rule 			
<pre>e_dict_map(tab_to_dict(tab_ "city", "popu",)</pre>	parse_csv("city,pop\nsh,2000\nn	j,800"), "city", "pop"),	
 Result 			
k1:v1 city:nj popu:800			
Example 2Raw log			
k1:v1 city:js,nj			
 Transformation rule 			
<pre>e_dict_map(tab_to_dict(tab_parse_cs ["province", "pop",), "city", "popu",))</pre>	v("province,city,pop\nsh,sh,2000 "city"],)\njs,nj,800"),	
 Result 			
k1:v1 city:js,nj popu:800			

4.5.8.7.16. Resource functions

This topic describes the syntax and parameters of resource functions. This topic also provides examples on how to use the functions.

Functions

() Important When you call the following resource functions, you must configure the Advanced preview mode to pull the data that you require.

Function	Description
res_local	Pulls the values of advanced parameters from the current data transformation job.
res_rds_mysql	Pulls data from a specified table in a database that is created on an ApsaraDB RDS for MySQL instance or obtains the execution result of an SQL statement. The data and result can be updated at regular intervals.
res_log_logstore_pull	Pulls data from another Logstore when you transform data in a Logstore. You can pull data in a continuous manner.
res_oss_file	Pulls data from an object in a specified Object Storage Service (OSS) bucket. The data can be updated at regular intervals.

res_local

The res_local function pulls the values of advanced parameters from the current data transformation job.

- Syntax
 - res_local(param, default=None, type="auto")
- Parameters

Parameter	Туре	Required	Description
param	String	Yes	The key that is specified in Advanced Parameter Settings for the current data transformation job.
default	String	No	The value that is returned if the key specified for the param parameter does not exist. Default value: None.

type String No	 The format of the output data. Valid values: auto: Raw data is converted to a JSON string. If the conversion fails, the raw data is returned. This is the default value. JSON: Raw data is converted to a JSON string. If the conversion fails, the value of the default parameter is returned. raw: Raw data is returned.
----------------	--

Response

A JSON string or raw data is returned based on the parameter settings.

Successful conversions

Raw data	Return value	Return value type
1	1	Integer
1.2	1.2	Float
true	True	Boolean
false	False	Boolean
"123"	123	String
null	None	None
["v1", "v2", "v3"]	["v1", "v2", "v3"]	List
["v1", 3, 4.0]	["v1", 3, 4.0]	List
{"v1": 100, "v2": "good"}	{"v1": 100, "v2": "good"}	List
{"v1": {"v11": 100, "v2": 200}, "v3": "good"}	{"v1": {"v11": 100, "v2": 200}, "v3": "good"}	List

Failed conversions

The following table provides some examples of failed conversions. The following raw data fails to be converted to JSON strings, and the raw data is returned as strings.

Raw data	Return value	Description
(1,2,3)	"(1,2,3)"	Tuples are not supported. Lists must be used.
True	"True"	A value of the Boolean data type can only be true or false. The values must be in lowercase.
{1: 2, 3: 4}	"{1: 2, 3: 4}"	A dictionary key can only be a string.

Example

Obtain the key specified in Advanced Parameter Settings and assign the value of the obtained key to the local parameter.

In Advanced Parameter Settings, the key is endpoint , and the value is hangzhou .

• Raw log

content: 1

• Transformation rule

e_set("local", res_local('endpoint'))

• Result

content: 1 local: hangzhou

res_rds_mysql

The res_rds_mysql function pulls data from a specified table in a database that is created on an ApsaraDB RDS for MySQL instance or obtains the execution result of an SQL statement. Log Service allows you to pull data by using the following methods:

! Important

- If you use the res_rds_mysql function to pull data from a database that is created on an ApsaraDB RDS for MySQL instance, you must create a whitelist on the instance and add 0.0.0.0 to the whitelist. This allows access to the database from all IP addresses. However, this may create risks for the database. If you want to add the IP address of Log Service to the whitelist, you must submit a ticket.
- Log Service can access a database that is created on an ApsaraDB RDS for MySQL instance by using either a public or internal endpoint of the instance. If an internal endpoint is used, you must configure the advanced parameters.

Pull all data only once

When you run a data transformation job for the first time, Log Service pulls all data from a specified table and then no longer pulls data. If your database is not updated, we recommend that you use this method.

Pull all data at regular intervals

When you run a data transformation job, Log Service pulls all data from a specified table at regular intervals. This way, Log Service can synchronize data with your database in a timely manner. However, this method requires a long period of time. If the data volume of your database is less than or equal to 2 GB and the value of the **refresh_interval** parameter is greater than or equal to 300 seconds, we recommend that you use this method.

Pull incremental data at regular intervals

When you run a data transformation job, Log Service pulls only incremental data based on the timestamp field of a specified database. If you use this method, Log Service pulls only the newly added data. This method is efficient. You can set the **refresh_interval** parameter to 1 second to synchronize data within seconds. If the data volume of your database is large, your data is frequently updated, or you require data to be pulled in a timely manner, we recommend that you use this method.

Syntax

res_rds_mysql(address="The address of the database from which data is pulled", username="The username used to connect to the database", password="The password used to connect to the database", database="The name of the database", table=None, fields=None, fields=None, refresh_interval=0, base_retry_back_off=1, max_retry_back_off=60, primary_keys=None, use_ssl=false, update_time_key=None, deleted_flag_key=None)

③ Note You can also use the res_rds_mysql function to pull data from a database that is created in an AnalyticDB for MySQL or PolarDB for MySQL cluster. In these scenarios, you need to only replace the address, username, password, and name of the database in the transformation rule with the actual values.

• Parameters

Parameter	Туре	Required	Description
address	String	Yes	The endpoint or IP address of the database to which you want to connect. If the port number is not 3306, specify a value in the IP address:Port format. For more information, see View and change the internal and public endpoints and port numbers of an ApsaraDB RDS for SQL Server instance.
username	String	Yes	The username that is used to connect to the database.
password	String	Yes	The password that is used to connect to the database.
database	String	Yes	The name of the database to which you want to connect.
table	String	Yes	The name of the table from which you want to pull data. If the sql parameter is configured, this parameter is not required.
sql	String	Yes	The SQL SELECT statement that is used to query data. If the table parameter is configured, this parameter is not required.
fields	String List	No	The string list or string alias list. If you do not configure this parameter, all columns returned for the sql or table parameter are used. For example, if you want to rename the name column in the ["user_id", "province", "city", "name", "age"] list to user_name, you must set the fields parameter to ["user_id", "province", "city", ("name", "user_name"), ("nickname", "nick_name"), "age"]. ③ Note If you configure the sql, table, and fields parameters together, the SQL statement in the sql parameter is executed. The table and fields parameters do not take effect.
fetch_include_data	String	No	 The field whitelist. Logs whose fields match the fetch include data parameter are retained. Logs whose fields do not match this parameter are discarded. If you do not configure this parameter or set this parameter toNone, the field whitelist feature is disabled. If you set this parameter to a specific field and field value, the logs that contain the field and field value are retained.
fetch_exclude_data	String	No	 The field blacklist. Logs whose fields match the fetch_exclude_data parameter are discarded. Logs whose fields do not match this parameter are retained. If you do not configure this parameter or set this parameter toNone, the field blacklist feature is disabled. If you set this parameter to a specific field and field value, the logs that contain the field and field value are discarded. Note If you configure both the fetch_include_data and fetch_exclude_data parameters, data is pulled first based on the setting of the fetch_include_data parameter and then based on the setting of the fetch_exclude_data parameter.
refresh_interval	Numeric string or number	No	The interval at which data is pulled from ApsaraDB RDS for MySQL. Unit: seconds. Default value: 0. This value indicates that all data is pulled only once.
base_retry_back_off	Number	No	The interval at which the system attempts to pull data again after a pulling failure. Default value: 1. Unit: seconds.
max_retry_back_off	int	No	The maximum interval between two consecutive retries after a pulling failure. Default value: 60. Unit: seconds. We recommend that you use the default value.
primary_keys	String/List	No	The primary key. If you configure this parameter, data in the table is saved to memory as a dictionary in the Key:Value format. A key in the dictionary is the value of the primary_keys parameter. A value in the dictionary is an entire row of data in the table. Image: The table of the table Image: Table of table. Image: Table of table. Image: Table of table of table of table of table of table. Image: Table of table. Image: Table of table of table.

use_ssl	Boolean	No	Specifies whether to use the SSL protocol to connect to the ApsaraDB RDS for MySQL instance. Default value: false. Note If SSL encryption is enabled on the ApsaraDB RDS for MySQL instance, Log Service connects to the instance over SSL. However, the server certificate is not verified. The server certificate cannot be used to establish connections.
update_time_key	String	No	The time field that is used to pull incremental data. If you do not configure this parameter, all data is pulled. For example, update_time in update_time_key="update_time" indicates the time field data update in the database. The time field supports the following data types: datetime, timestamp, integer, float, and decimal. Make sure that the values of the time field increase in chronological order. ③ Note Log Service pulls incremental data based on the time field. Make sure that an index is configured for this field in the table. If the index is not configured, a full table scan is performed. In addition, an error is reported, which indicates a failure to pull incremental data.
deleted_flag_key	String	No	The data that does not need to be transformed and is discarded when incremental data is pulled. For example, if the value of key in update_time_key="key" meets the following condition, the value is parsed as deleted data: • Boolean: true • Datetime and timestamp: not empty • Char and varchar: 1, true, t, yes, and y • Integer: non-zero ? Note • You must configure the deleted_flag_key parameter together with the update_time_key parameter. • If you configure the update_time_key parameter but do not configure the deleted_flag_key parameter, no data is discarded when incremental data is pulled.
connector	String	No	The connector that is used to remotely connect to the database. Valid values: mysql and pgsql. Default value: mysql.

Response

A table that contains multiple columns is returned. The columns are defined by the **fields** parameter.

• Error handling

If an error occurs when data is pulled, the error is reported, but the data transformation job continues to run. Retries are performed based on the value of the **base_retry_back_off** parameter. For example, the first retry interval is 1 second and the first retry fails. The second retry interval is twice the length of the first one. The process continues until the interval reaches the value of the **max_retry_back_off** parameter. If the error persists, retries are performed based on the value of the **max_retry_back_off** parameter. If the error value, which is 1 second.

- Examples
 - Pull all data

• Example 1: Pull data from the test_table table in the test_db database at 300-second intervals.

res_rds_mysql(

```
address="rm-uf6wjk5****.mysql.rds.aliyuncs.com",
username="test_username",
password="****",
database="test_db",
table="test_table",
refresh_interval=300,
```

)

• Example 2: Pull data from the test_table table, excluding the data records whose status value is delete.

- res_rds_mysql(
 - address="rm-uf6wjk5****.mysql.rds.aliyuncs.com", username="test_username", password="****", database="test_db", table="test_table", refresh_interval=300, fetch_exclude_data="'status':'delete'",

• Example 3: Pull the data records whose **status** value is exit from the test_table table.

```
res_rds_mysql(
    address="rm-uf6wjk5****.mysql.rds.aliyuncs.com",
    username="test_username",
    password="***",
    database="test_db",
    table="test_table",
    refresh_interval=300,
    fetch_include_data="'status':'exit'",
)
```

• Example 4: Pull the data records whose status value is exit from the test_table table, excluding the data records whose name value is aliyun.

```
res rds mysgl(
   address="rm-uf6wjk5****.mysql.rds.aliyuncs.com",
   username="test_username",
   password="****",
   database="test db",
   table="test_table",
   refresh_interval=300,
   fetch_include_data="'status':'exit'",
   fetch_exclude_data="'name':'aliyun'",
```

• Example 5: Use the pgsql connector to connect to a Hologres database and pull data from the test_table table.

```
res_rds_mysql(
    address="hgpostcn-cn-****-cn-hangzhou.hologres.aliyuncs.com:80",
   username="test_username",
password="****",
    database="aliyun",
   table="test_table",
    connector="pgsql",
)
```

)

Example 6: Pull data by using the primary_keys parameter.

If you configure the **primary_keys** parameter, its value is extracted as a key. The data pulled from the table is saved to memory in the **{"10001":{"userid":"10001", city_name":"beijing", city_number":"12345"}}** format. In this case, data is pulled at a high speed. If you want to pull a large volume of data, we recommend that you use this method. If you do not configure the primary_keys parameter, the function traverses the table by row. Then, the function pulls data and saves the data to memory in the **[{"userid":"10001", city_name":"beijing", city_number":"12345"}}** format. In this case, data is pulled at a low speed, but only a small portion of memory is occupied. If you want to pull a small volume of data, we recommend that you use this method.

Table

userid	city_name	city_number
10001	beijing	12345

Raw log

```
# Data Record 1
   userid:10001
   gdp:1000
   # Data Record 2
   userid:10002
   gdp:800

    Transformation rule

   e_table_map(
       rds_mysql(
    address="rm-uf6wjk5****.mysql.rds.aliyuncs.com",
            username="test_username",
           password="****",
           database="test db",
           table="test table".
           primary_keys="userid",
       ),
        "userid",
       ["city_name", "city_number"],
   )

    Result

   # Data Record 1
   userid:10001
   gdp:1000
   city_name: beijing
   city number:12345
   # Data Record 2
   userid:10002
   gdp:800
```

Pull incremental data

Example 1: Pull incremental data.

- () Note You can pull incremental data from a table only if the following conditions are met:
 - The table has a unique primary key and a time field, such as the item_id and update_time fields.
 - The primary_keys, refresh_interval, and update_time_key parameters are configured.

Table

item_id	item_name	price
1001	Orange	10
1002	Apple	12
1003	Mango	16

Raw log

#	Data	Re	ecord	1
it	em_io	1:	1001	
to	otal:	1(00	
#	Data	Re	ecord	2

item_id: 1002 total: 200

Data Record 3
item_id: 1003
total: 300

Transformation rule

e_table_map(
 res_rds_mysql(
 address="rm-uf6wjk5****.mysql.rds.aliyuncs.com",
 username="test_username",
 password="****",
 database="test_db",
 table="test_table",
 primary_key="item_id",
 refresh_interval=1,
 update_time_key="update_time",
),

"item_id",

["item_name", "price"],

) • Result

Data Record 1
item_id: 1001
total: 100
item_name: Orange
price:10

Data Record 2
item_id: 1002
total: 200
item_name: Apple
price:12

Data Record 3
item_id: 1003
total: 300
item_name: Mango
price:16

• Example 2: Configure the **deleted_flag_key** parameter to discard specified data when incremental data is pulled.

```
    Table
```

item_id	item_name	price	update_time	Is_deleted
1001	Orange	10	1603856138	False
1002	Apple	12	1603856140	False
1003	Mango	16	1603856150	False

Raw log

Data Record 1
item_id: 1001
total: 100
Data Record 2
item_id: 1002
total: 200
Data Record 3
item id: 1003

total: 300
Transformation rule

e_table_map(

res_rds_mysql(
 address="rm-uf6wjk5****.mysql.rds.aliyuncs.com",
 username="test_username",
 password="****",
 database="test_db",
 table="test_table",
 primary_key="item_id",
 refresh_interval=1,
 update_time_key="update_time",
 deleted_flag_key="is_deleted",
),
"item_id",

["item_name", "price"],

Result

The res_rds_mysq function pulls three data records from the table to the memory of the server on which Log Service runs. These data records are compared with the existing data records in the source Logstore to check whether the records match. If you want to discard the data record whose item_id is **1001**, find the data record whose item_id is **1001** in the table and change the value of the **Is_deleted** field to **true**. This way, the data record **1001** is discarded the next time that the in-memory dimension table is updated.

Data Record 2
item_id: 1002
total: 200
item_name: Apple
price:12
Data Record 3

item_id: 1003
total: 300
item_name: Mango
price:1

res_log_logstore_pull

The res_log_logstore_pull function pulls data from another Logstore when you transform data in a Logstore.

Syntax

res_log_logstore_pull(endpoint, ak_id, ak_secret, project, logstore, fields, from_time="begin", to_time=None, fetch_include_data=None, fetch_exclude_data=None, primary_keys=None, fetch_interval=2, delete_data=None, base_retry_back_off=1, max_retry_back_off=60, ttl=None, role_arn=None)

Parameters

Parameter	Туре	Required	Description
endpoint	String	Yes	The endpoint. By default, an HTTPS endpoint is used. You can also use an HTTP endpoint. In special cases, you may need to use a port other than port 80 or 443.
ak_id	String	Yes	The AccessKey ID of your Apsara Stack tenant account. To ensure data security, we recommend that you configure this parameter in Advanced Parameter Settings .
ak_secret	String	Yes	The AccessKey secret of your Apsara Stack tenant account. To ensure data security, we recommend that you configure this parameter in Advanced Parameter Settings .
project	String	Yes	The name of the project from which you want to pull data.
logstore	String	Yes	The name of the Logstore from which you want to pull data.

fields	String List	Yes	The string list or string alias list. If a log does not contain a specified field, the value of this field is an empty string. For example, if you want to rename the name column in the ['user_id' , 'province' , 'city' , "name' , " age'] list to user_name , you must set this parameter to ['user_id' , 'province' , 'city' , ('name' , 'user_name'), ("nickname' , "nick_name'), "age'].
from_time	String	No	 The server time when the first data pulling from the Logstore starts. Default value: begin. This value indicates that Log Service starts to pull data from the first log. The following time formats are supported: UNIX timestamp. Time string. Custom string, such as begin or end. Expression: the time that is returned by the dt_function. For example, the function dt_totimestamp(dt_truncate(dt_today(tz="Asia/Shanghai"), day=op_neg(-1))) returns the start time of data pulling, which is one day before the current time. If the current time is 2019-5-5 10:10:10 (UTC+8), the returned time is 2019-5-4 10:10:10 (UTC+8).
to_time	String	No	The server time when the first data pulling from the Logstore ends. Default value: None. This value indicates that Log Service stops pulling data at the last log. The following time formats are supported: • UNIX timestamp. • Time string. • Custom string, such as begin or end. • Expression: the time that is returned by the dt_function. If you do not configure this parameter or set this parameter toNone, data is pulled from the latest logs in a continuous manner. () Note If you set this parameter to a point in time that is later than the current time, only the existing data in the Logstore is pulled. New data is not pulled.
fetch_include_data	String	No	 The field whitelist. Logs whose fields match the fetch_include_data parameter are retained. Logs whose fields do not match this parameter are discarded. If you do not configure this parameter or set this parameter toNone, the field whitelist feature is disabled. If you set this parameter to a specific field and field value, the logs that contain the field and field value are retained.
fetch_exclude_data	String	No	The field blacklist. Logs whose fields match the fetch_exclude_data parameter are discarded. Logs whose fields do not match this parameter are retained. • If you do not configure this parameter or set this parameter to None , the field blacklist feature is disabled. • If you set this parameter to a specific field and field value, the logs that contain the field and field value are discarded. • ONDE If you configure both the fetch_include_data and fetch_exclude_data parameters, data is pulled first based on the setting of the fetch_include_data parameter.
primary_keys	String list	No	 The list of primary key fields that are used to maintain a table. If you change the name of a primary key field by using the fields parameter, you must use the new name to specify the primary key field for this parameter. Note The value of the primary_keys parameter can contain only the single-value strings that are specified in the value of the fields parameter. This parameter is valid when only one shard exists in the Logstore from which data is pulled.
fetch_interval	Int	No	The interval between two consecutive data pulling requests when data is pulled in a continuous manner. Default value: 2. Unit: seconds. The value must be greater than or equal to 1 second.
delete_data	String	No	The operation to delete data from the table. Data records that meet specified conditions and contain the value of <code>primary_keys</code> are deleted.
base_retry_back_off	Number	No	The interval at which the system attempts to pull data again after a pulling failure. Default value: 1. Unit: seconds.
max_retry_back_off	Int	No	The maximum interval between two consecutive retries after a pulling failure. Default value: 60. Unit: seconds. We recommend that you use the default value.
ttl	Int	No	The number of seconds that is used to determine the range for data pulling in a continuous manner. Data pulling starts when log data is generated and ends ttl seconds after the time that the log data is generated. Unit: seconds. Default value: None. This value indicates that all log data is pulled.
role_arn	String	No	The Alibaba Cloud Resource Name (ARN) of the RAM role that is used. The RAM role must have the read permissions on the Logstore from which data is pulled. In the RAM console, you can view the ARN of a RAM role in the Basic Information section on the details page of the role. Example: acs:ram::1379******44:role/role-a

Response

A table that contains multiple columns is returned.

Error handling

If an error occurs when data is pulled, the error is reported, but the data transformation job continues to run. Retries are performed based on the value of the **base_retry_back_off** parameter. For example, the first retry interval is 1 second and the first retry fails. The second retry interval is twice the length of the first one. The process continues until the interval reaches the value of the **max_retry_back_off** parameter. If the error persists, retries are performed based on the value of the **max_retry_back_off** parameter. If the error value, which is 1 second.

- Examples
- In this example, the data of fields key1 and key2 is pulled from the test_logstore Logstore of the test_project project. Data pulling starts when log
 data is written to the Logstore. Data pulling ends when the data write operation is complete. The data is pulled only once.

```
res_log_logstore_pull(
    "cn-hangzhou.log.aliyuncs.com",
    "LT****Gw",
    "ab****uu",
    "test_project",
    "test_logstore",
    ["key1", "key2"],
    from_time="begin",
    to_time="end",
```

)

- In this example, the data of fields key1 and key2 is pulled from the test_logstore Logstore of the test_project project. Data pulling starts when log
 data is written to the Logstore. Data pulling ends when the data write operation is complete. The data is pulled at 30-second intervals in a
 continuous manner.
 - res_log_logstore_pull(
 "cn-hangzhou.log.aliyuncs.com",
 "LT****Gw",
 "ab***tuu",
 "test_project",
 "test_logstore",
 ["keyl", "key2"],
 from_time="Negin",
 to_time=None,
 fetch_interval=30,
- In this example, a blacklist is configured to skip the data records that contain key1:value1 when data is pulled from a Logstore.
 - res_log_logstore_pull(
 "cn-hangzhou.log.aliyuncs.com",
 "LT****Gw",
 "ab****uu",
 "test_project",
 "test_logstore",
 ["key1", "key2"],
 from_time="begin",
 to_time=None,
 fetch_interval=30,
 fetch_exclude_data="key1:valuel",

• In this example, a whitelist is configured to pull the data records that contain **key1:value1** from a Logstore.

- res_log_logstore_pull(
 "cn-hangxhou.log.aliyuncs.com",
 "LT****Gw",
 "ab****uu",
 "test_project",
 "test_logstore",
 ["key1", "key2"],
 from_time="begin",
 to_time=None,
 fetch_interval=30,
 fetch_include_data="key1:value1",
-)

)

In this example, the data of fields key1 and key2 is pulled from the test_logstore Logstore of the test_project project. Data pulling starts when log
data is generated and ends 40,000,000 seconds after the time that the log data is generated.

```
res_log_logstore_pull(
    "cn-hangzhou.log.aliyuncs.com",
    "LTAI*****Cajvr",
    "q00Tp*****j39",
    "test_project",
    "test_logstore",
    fields=["key1","key2"],
    ttl="40000000"
)
```

Cloud Defined Storage

 In this example, the data of fields key1 and key2 is pulled from the test-logstore Logstore of the project-test1 project. The service-linked role of Log Service is used for authorization. Data pulling starts when log data is written to the Logstore. Data pulling ends when the data write operation is complete. The data is pulled only once.

es_	log_logstore_pull(
	"pub-cn-hangzhou-staging-intranet.log.aliyuncs.com",
	"",
	"",
	"project-test1",
	"test-logstore",
	["key1", "key2"],
	from_time="2022-7-27 10:10:10 8:00",
	to_time="2022-7-27 14:30:10 8:00",
	role_arn="acs:ram::***:role/aliyunservicerolefors1saudit"

-)
- In this example, the data of fields key1 and key2 is pulled from the test-logstore Logstore of the project-test1 project. A default role is used for authorization. Data pulling starts when log data is written to the Logstore. Data pulling ends when the data write operation is complete. The data is pulled only once.

es_log_logstore_pull(
"cn-chengdu.log.aliyuncs.com",
"",
"",
"project-test1",
"test-logstore",
["key1", "key2"],
from_time="2022-7-21 10:10:10 8:00",
to_time="2022-7-21 10:30:10 8:00",
role_arn="acs:ram::***:role/aliyunlogetlrole"

res_oss_file

The res_oss_file function pulls data from an object in a specified OSS bucket. The data can be updated at regular intervals.

• Syntax

res_oss_file(endpoint, ak_id, ak_key, bucket, file, format='text', change_detect_interval=0, base_retry_back_off=1, max_retry_back_off=60, e ncoding='utf8', error='ignore')

• Parameters

Parameter	Туре	Required	Description
endpoint	String	Yes	The endpoint of the OSS bucket. By default, an HTTPS endpoint is used. You can also use an HTTP endpoint. In special cases, you may need to use a port other than port 80 or 443.
ak_id	String	Yes	The AccessKey ID of your Apsara Stack tenant account. To ensure data security, we recommend that you configure this parameter in Advanced Parameter Settings .
ak_key	String	Yes	The AccessKey secret of your Apsara Stack tenant account. To ensure data security, we recommend that you configure this parameter in Advanced Parameter Settings .
bucket	String	Yes	The name of the OSS bucket from which you want to pull data.
file	String	Yes	The path to the object from which you want to pull data. Example: test/data.txt. Do not enter a forward slash (/) at the start of the path.
format	String	Yes	The format of the output file. Valid values: • Text: text format • Binary: byte stream format
change_detect_interval	String	No	The interval at which Log Service pulls the object data from OSS. Unit: seconds. The system checks whether the object is updated when data is pulled. If the object is updated, the incremental data is pulled. Default value: 0. This value indicates that no incremental data is pulled. All data is pulled only once when the function is called.
base_retry_back_off	Number	No	The interval at which the system attempts to pull data again after a pulling failure. Default value: 1. Unit: seconds.
max_retry_back_off	int	No	The maximum interval between two consecutive retries after a pulling failure. Default value: 60. Unit: seconds. We recommend that you use the default value.
encoding	String	No	The encoding format. If you set the format parameter to Text, this parameter is automatically set to utf8.
error	String	No	 The method that is used to handle errors. This parameter is valid only when the UnicodeError message is reported. Valid values: ignore: The system skips the data of an invalid format and continues to encode data. xmlcharrefreplace: The system uses appropriate XML character references to replace the characters that cannot be encoded.
decompress	String	No	 Specifies whether to decompress the object. Valid values: None (default): The object is not decompressed. gzip: The object is decompressed by using gzip.

Response

Object data is returned in the byte stream or text format.

Error handling

If an error occurs when data is pulled, the error is reported, but the data transformation job continues to run. Retries are performed based on the value of the **base_retry_back_off** parameter. For example, the first retry interval is 1 second and the first retry fails. The second retry interval is twice the length of the first one. The process continues until the interval reaches the value of the **max_retry_back_off** parameter. If the error persists, retries are performed based on the value of the **max_retry_back_off** parameter. If the error value, which is 1 second.

- Examples
- Example 1: Pull JSON data from OSS.
- JSON data

```
"users": [
  {
      "name": "user1".
      "login_historys": [
       {
          "date": "2019-10-10 0:0:0",
         "login_ip": "203.0.113.10"
        },
       {
          "date": "2019-10-10 1:0:0",
          "login_ip": "203.0.113.10"
       }
      ]
  },
  {
      "name": "user2",
      "login_historys": [
       {
          "date": "2019-10-11 0:0:0",
          "login_ip": "203.0.113.20"
        },
        {
          "date": "2019-10-11 1:0:0",
          "login_ip": "203.0.113.30"
        },
        {
          "date": "2019-10-11 1:1:0",
          "login_ip": "203.0.113.50"
        }
     1
  }
]
```

Raw log

content: 123

```
    Transformation rule
```

```
e_set(
    "json_parse",
    json_parse(
        res_oss_file(
            endpoint="http://oss-cn-hangzhou.aliyuncs.com",
            ak_id="LT****Gw",
            ak_key="ab****uu",
            bucket="log-etl-staging",
            file="testjson.json",
            ),
      ),
      ),
```

```
    Result
```

```
content: 123
prjson_parse:
  "users": [
   {
        "name": "user1",
       "login_historys": [
         {
           "date": "2019-10-10 0:0:0",
"login_ip": "203.0.113.10"
          },
         {
            "date": "2019-10-10 1:0:0",
           "login_ip": "203.0.113.10"
         }
       ]
    },
    {
        "name": "user2",
        "login_historys": [
         {
            "date": "2019-10-11 0:0:0",
           "login_ip": "203.0.113.20"
          },
         {
"date": "2019-10-11 1:0:0",
           "login_ip": "203.0.113.30"
          },
         {
    "date": "2019-10-11 1:1:0",
           "login_ip": "203.0.113.50"
          }
       ]
   }
]
}'
```

• Example 2: Pull text content from OSS.

```
    Text content
```

- Test bytes

 Raw log
- i la li log

```
content: 123
```

Transformation rule

```
e_set(
    "test_txt",
    res_oss_file(
        endpoint="http://oss-cn-hangzhou.aliyuncs.com",
        ak_id="LT***Gw",
        ak_key="ab****uu",
        bucket="log-etl-staging",
        file="test.txt",
        ),
    )
```

Result

content: 123 test_txt: Test bytes

- Example 3: Pull data from a compressed OSS object and decompress the object.
 - Content of the compressed object

```
Test bytes\nupdate\n123
```

```
    Raw log
```

content:123 Transformation rule

```
e_set(
   "text",
   res_oss_file(
      endpoint="http://oss-cn-hangzhou.aliyuncs.com",
       ak_id="LT***Gw",
      ak_key="ab****uu",
      bucket="log-etl-staging",
      file="test.txt.gz",
      format="binary",
       change_detect_interval=30,
       decompress="gzip",
   ),
```

) Result

```
content:123
text: Test bytes\nupdate\n123
```

• Example 4: Access an object in an OSS bucket whose ACL is public-read-write. No AccessKey pairs are used.

```
    Content of the compressed object

   Test bytes

    Raw log

   content:123

    Transformation rule

   e_set(
       "test_txt",
       res_oss_file(
            endpoint="http://oss-cn-hangzhou.aliyuncs.com",
           bucket="log-etl-staging",
            file="test.txt",
       ),
   )
```

Result

content: 123 test_txt: Test bytes

4.5.8.7.17. Parsing functions

This topic describes the syntax and parameters of User-Agent parsing functions. This topic also provides examples on how to use the functions. Functions

г	u	r	C	L	1	υ	r	ļ

Function	Description
ua_parse_device	Parses User-Agent and returns the device information.
ua_parse_os	Parses User-Agent and returns the operating system information.
ua_parse_agent	Parses User-Agent and returns the browser information.
ua_parse_all	Parses User-Agent and returns all information.

O Note The User-Agent parsing functions delete fields whose parsed values are None. For example, if the device information that is obtained is ('brand': None, 'family': 'Other', 'model': None), the brand and model fields are deleted and the final parsing result is ('family': 'Other'} .

ua_parse_device

The ua_parse_device function parses User-Agent and returns the device information.

Syntax

ua_parse_device(value)

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The User-Agent string that you want to parse. Example: ua_parse_device(v("http_user_agent")) .

Response

A JSON-formatted data set is returned.

- Example
- Raw log

http_user_agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/192.168.0.0 Safari/537.36

Transformation rule

e_set("new_column",ua_parse_device(v("http_user_agent")))

• Result

http_user_agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/192.168.0.0 Safari/537.36 new_column:{'family': 'Other'}

ua_parse_os

The ua_parse_os function parses User-Agent and returns the operating system information.

Syntax

ua_parse_os(value)

• Parameters

Parameter	Туре	Required	Description
value	String	Yes	The User-Agent string that you want to parse. Example: ua_parse_os(v("http_user_agent")) .

Response

A JSON-formatted data set is returned.

- Example
- Raw log

http_user_agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/192.168.0.0 Safari/537.36

• Transformation rule

e_set("new_column",ua_parse_os(v("http_user_agent")))

Result

http_user_agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/192.168.0.0 Safari/537.36 new_column:{'family': 'Mac OS X',

'major': '10', 'minor': '9', 'patch': '4'}

ua_parse_agent

The ua_parse_agent function parses User-Agent and returns the browser information.

• Syntax

ua_parse_agent(value)

Parameters

Parameter	Туре	Required	Description
value	String	Yes	The User-Agent string that you want to parse. Example: ua_parse_agent(v("http_user_agent")) .

- Response
- A JSON-formatted data set is returned.
- Example
- Raw log

http_user_agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/192.168.0.0 Safari/537.36

Transformation rule

e_set("new_column",ua_parse_agent(v("http_user_agent")))

• Result

http_user_agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/192.168.0.0 Safari/537.36 new_columa:{'family': 'Chrome', 'major': '192', 'minor': '168', 'patch': '0'}

ua_parse_all

The ua_parse_all function parses User-Agent and returns all information.

• Syntax

ua_parse_all(value)

• Parameters

•

Parameter	Туре	Required	Description
value	String	Yes	The User-Agent string that you want to parse. Example: ua_parse_all(v("http_user_agent")) .
Response			
A JSON-formatted data set is re	eturned.		
Example			
 Raw log 			
http_user_agent:Mozilla/5.	.0 (Macintosh; Intel Mac OS X	10_9_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/192.168.0.0 Safari/537.36
 Transformation rule 			
e_set("new_column",ua_pars	<pre>se_all(v("http_user_agent")))</pre>		
• Result			
<pre>http_user_agent: Mozilla/5 new_column: { "user_agent": { "family": "Chrome", "major": "192", "minor": "168", "patch": "0" }, "os": { "family": "Mac OS X", "major": "10", "minor": "9", "patch": "4" }, "device": { "family": "Mac", "brand": "Apple", "model": "Mac" } }</pre>	5.0 (Macintosh; Intel Mac OS	X 10_9_4) AppleWebKit/537.36	(KHTML, like Gecko) Chrome/192.168.0.0 Safari/537.36

4.5.8.8. General reference

4.5.8.8.1. Standard encoding formats

This topic describes the standard encoding formats that are supported by Log Service.

Encoding format	Alias	Language
ascii	646 and us-ascii	English
big5	big5-tw and csbig5	Traditional Chinese
big5hkscs	big5-hkscs and hkscs	Traditional Chinese
cp037	IBM037 and IBM039	English
cp273	273, IBM273, and csIBM273	German
cp424	EBCDIC-CP-HE and IBM424	Hebrew
cp437	437 and IBM437	English
cp500	EBCDIC-CP-BE, EBCDIC-CP-CH, and IBM500	Western European languages
cp720	None	Arabic
cp737	None	Greek
cp775	None	IBM775
cp850	850 and IBM850	Western European languages
cp852	852 and IBM852	Central and Eastern European languages
cp855	855 and IBM855	Bulgarian, Belarusian, Macedonian, Russian, and Serbian
cp856	None	Hebrew
cp857	857 and IBM857	Turkish
cp858	858 and IBM858	Western European languages
cp860	860 and IBM860	Portuguese
cp861	861, CP-IS, and IBM861	Icelandic
cp862	862 and IBM862	Hebrew
cp863	863 and IBM863	Canadian

cp864	IBM864	Arabic
cp865	865 and IBM865	Danish and Norwegian
cp866	866 and IBM866	Russian
cp869	869, CP-GR, and IBM869	Greek
cp874	None	Thai
cp875	None	Greek
cp932	932, ms932, mskanji, and ms-kanji	Japanese
cp949	949, ms949, and uhc	Korean
cp950	950 and ms950	Traditional Chinese
cp1006	None	Urdu
cp1026	ibm1026	Turkish
cp1125	1125, ibm1125, cp866u, and ruscii	Ukrainian
cp1140	ibm1140	Western European languages
cp1250	windows-1250	Central and Eastern European languages
cp1251	windows-1251	Bulgarian, Belarusian, Macedonian, Russian, and Serbian
cp1252	windows-1252	Western European languages
cp1253	windows-1253	Greek
cp1254	windows-1254	Turkish
cp1255	windows-1255	Hebrew
cp1256	windows-1256	Arabic
cp1257	windows-1257	Baltic languages
cp1258	windows-1258	Vietnamese
cp65001	None	Windows only: Windows UTF-8 (CP_UTF8)
euc_jp	eucjp, ujis, and u-jis	Japanese
euc_jis_2004	jisx0213 and eucjis2004	Japanese
euc_jisx0213	eucjisx0213	Japanese
euc_kr	euckr, korean, ksc5601, ks_c-5601, ks_c-5601-1987, ksx1001, and ks_x-1001	Korean
gb2312	chinese, csiso58gb231280, euc-cn, euccn, eucgb2312-cn, gb2312-1980, gb2312-80, and iso-ir- 58	Simplified Chinese
gbk	936, cp936, and ms936	CJK Unified Ideographs
gb18030	gb18030-2000	CJK Unified Ideographs
hz	hzgb, hz-gb, and hz-gb-2312	Simplified Chinese
iso2022_jp	csiso2022jp, iso2022jp, and iso-2022-jp	Japanese
iso2022_jp_1	iso2022jp-1 and iso-2022-jp-1	Japanese
iso2022_jp_2	iso2022jp-2 and iso-2022-jp-2	Japanese, Korean, Simplified Chinese, Western European languages, and Greek
iso2022_jp_2004	iso2022jp-2004 and iso-2022-jp-2004	Japanese
iso2022_jp_3	iso2022jp-3 and iso-2022-jp-3	Japanese
iso2022_jp_ext	iso2022jp-ext and iso-2022-jp-ext	Japanese
iso2022_kr	csiso2022kr, iso2022kr, and iso-2022-kr	Korean
latin_1	iso-8859-1, iso8859-1, 8859, cp819, latin, latin1, and L1	Western European languages
iso8859_2	iso-8859-2, latin2, and L2	Central and Eastern European languages
iso8859_3	iso-8859-3, latin3, and L3	Esperanto and Maltese
iso8859_4	iso-8859-4, Latin4, and L4	Baltic languages
iso8859_5	iso-8859-5 and cyrillic	Bulgarian, Belarusian, Macedonian, Russian, and Serbian
iso8859_6	iso-8859-6 and arabic	Arabic

User Guide-Log Service

iso8859_7	iso-8859-7, greek, and greek8	Greek
iso8859_8	iso-8859-8 and hebrew	Hebrew
iso8859_9	iso-8859-9, Latin5, and L5	Turkish
iso8859_10	iso-8859-10, Latin6, and L6	Nordic languages
iso8859_11	iso-8859-11 and thai	Thai
iso8859_13	iso-8859-13, latin7, and L7	Baltic languages
iso8859_14	iso-8859-14, Latin8, and L8	Celtic
iso8859_15	iso-8859-15, latin9, and L9	Western European languages
iso8859_16	ISO-8859-16, Latin10, and L10	Southeast European languages
johab	cp1361 and ms1361	Korean
koi8_r	None	Russian
koi8_t	None	Tajik
koi8_u	None	Ukrainian
kz1048	kz_1048, strk1048_2002, and rk1048	Kazakh
mac_cyrillic	maccyrillic	Bulgarian, Belarusian, Macedonian, Russian, and Serbian
mac_greek	macgreek	Greek
mac_iceland	maciceland	Icelandic
mac_latin2	maclatin2 and maccentraleurope	Central and Eastern European languages
mac_roman	macroman and macintosh	Western European languages
mac_turkish	macturkish	Turkish
ptcp154	csptcp154, pt154, cp154, and cyrillic-asian	Kazakh
shift_jis	csshiftjis, shiftjis, sjis, and s_jis	Japanese
shift_jis_2004	shiftjis2004, sjis_2004, and sjis2004	Japanese
shift_jisx0213	shiftjisx0213, sjisx0213, and s_jisx0213	Japanese
utf_32	U32 and utf32	All languages
utf_32_be	UTF-32BE	All languages
utf_32_le	UTF-32LE	All languages
utf_16	U16 and utf16	All languages
utf_16_be	UTF-16BE	All languages
utf_16_le	UTF-16LE	All languages
utf_7	U7 and unicode-1-1-utf-7	All languages
utf_8	U8, UTF, and utf8	All languages
utf_8_sig	None	All languages

4.5.8.8.2. Query string syntax

Query strings are used in the domain-specific language (DSL) for Log Service to filter log data in an efficient manner and simplify condition matching. This topic describes the rules for specifying query strings.

Functions

The following table describes the functions that use query strings.

Category	Function	Scenario
Event check functions	e_search	Query strings are used to check whether the value of a field in an event meets specified conditions.
Descurse functions	res_log_logstore_pull	Query strings are used to configure a field blacklist or a field whitelist to filter data from a Logstore and return a table.
Resource functions	res_rds_mysql	Query strings are used to configure a field blacklist or a field whitelist to filter data from a specified table of an ApsaraDB RDS for MySQL database and return a table.
Event mapping functions	e_search_table_map and e_search_dict_map	Query strings are used to match key-value pairs in a dictionary.

Features

The following table lists the search features that support field search and full-text search.

Feature	Field search	Full-text search
Search for substrings	Supported	Supported
Search for strings by using wildcard characters, which support asterisks ($\stackrel{\star}{xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx$	Supported	Supported
Exact match	Supported	Not supported
Search for strings by using regular expressions	Supported	Not supported
Search for strings by comparing numeric ranges	Supported	Not supported
Search for strings by comparing numeric values	Supported	Not supported
Search for strings by using logical operators (AND, OR, and NOT), or a combination of these operators	Supported	Supported

Escape special characters

Special characters, such as asterisks (*) and backslashes (\), must be escaped in query strings.

- Escape special characters in a field name
- Field names cannot be enclosed in double quotation marks (""). Special characters in a field name must be escaped by using backslashes (\). Examples:
- $\ \(1+1\)\?: abc$. Special characters are escaped by using backslashes (\).
- __tag__\:__container_name__: abc . Special characters are escaped by using backslashes (\).
- Chinese field: abc : Chinese fields do not need to be escaped.
- "content": abc . In this example, the field name is invalid. The field name cannot be enclosed in double quotation marks ("").
- · Escape special characters in a field value
 - To query a field value that contains special characters such as quotation marks (") and backslashes (\), you must escape the special characters by using backslashes (\). Example: content: "abc\"xy\\z".

Note A field value must be enclosed in double quotation marks (""). You can use single quotation marks(") to enclose the string and double quotation marks ("") to enclose the field value. For example, e_search("domain: '/url/test.jsp'") is invalid, and e_search('domain: '/url/test.jsp'') is valid.

- To query a field value that contains special characters such as asterisks (*) and question marks (?), you must escape the special characters by using backslashes (\). If you do not escape the special characters by using backslashes (\), the special characters are used as wildcard characters for matching.
- To query a field value that contains only letters, digits, underscores (_), hyphens (-), asterisks (*), and question marks (?), you do not need to
 enclose the field value in double quotation marks (""). When other characters are used, you must enclose the field value in double quotation marks
 (""). Examples:
- status: "*\?()[]:=" . The field value is enclosed in double quotation marks (""). The asterisk (*) and question mark (?) are escaped by using backslashes (\). Characters other than the asterisk (*) and question mark (?) are not escaped in the field value.
- content: () []:= : The field value is invalid. The field value must be enclosed in double quotation marks ("").
- status: active*test and status: active\?test . The field values contain only letters, an asterisk (*) and a question mark (?). The field values do not need to be enclosed in double quotation marks (""). The asterisk (*) and question mark (?) in the field values are escaped by using backslashes (\).

Search for substrings

- Full-text search
 - Search for substrings in all fields.
- Syntax

e search('substring')

- Examples
 - e_search('"error"') : searches for a substring.
 - e_search('"active error"') : searches for a substring that contains a space.
 - e_search('active error') : searches for multiple substrings. The logical operator OR is used by default.
- Field search

Search for substrings in specific fields.

Syntax

e_search('...')

- Examples
- e_search('status: active') : searches for a substring.
- e_search('author: "john smith"') : searches for a substring that contains a space.

(2) Note e_search('field: active error') : searches for active in the field field or searches for error in all fields. In this example, the query string is equivalent to field:active OR "error".

Search for strings by using wildcard characters

An asterisk (*) specifies zero or multiple characters. A question mark (?) specifies one character or one wide character.

- Full-text search
- Search for substrings in all fields.
- Syntax

e_search('substring')

- Examples
 - e_search('active*test') . The asterisk (*) is used to match zero or multiple characters. The query string does not need to be enclosed in double quotation marks ("") because the query string contains only letters and an asterisk (*).
 - e_search('*error occurs') . The asterisk (*) is used to match zero or multiple characters. For example, the error occurs and critical error occurs strings can be matched.
 - e_search('active?good')
 The question mark (?) is used to match one character. The query string does not need to be enclosed in double quotation marks ("") because the query string contains only letters and a question mark (?).
 - e_search('ac*tive?good')
 The query string is used to perform an exact match by using an asterisk (*) and a question mark (?).
 - e_search('ac*tive??go*od')
 The query string is used to perform an exact match by using multiple asterisks (*) and question marks (?).

Field search

Search for substrings in specific fields.

Syntax

e_search('field name: substring')

- Examples
 - e_search('status: active*test') . The asterisk (*) is used to match zero or multiple characters.
 - e_search('status: active?good') : The question mark (?) is used to match one character.

Exact match

In exact match, the entire field value is matched.

Syntax

e_search('field name==string that must be exactly matched')

- Examples
- e_search('author== "john smith"') . The value of the author field must be john smith.
- e_search('status== ac*tive?good') . The query string contains wildcard characters and is used for exact match.

Search for strings by using regular expressions

Regular expressions are more efficient than wildcard characters in matching.

• Syntax

e_search('field name~=regular expression')

? Note

- Regular expressions may contain backslashes (\). We recommend that you use r to prevent the system from escaping the backslashes (\).
- By default, Log Service performs fuzzy match. To enable exact match, you must specify a regular expression that includes a caret (^) as a prefix and a dollar sign (\$) as a suffix.
- Examples
- e_search('status~= "\d+"') . The value of the **status** field contains digits.
- e_search('status~= "^\d+\$"') . The value of the **status** field is a number.

Search for strings by comparing numeric values or numeric ranges

- You can search for field values by comparing field values with specified numeric values or numeric ranges.
- Numeric value comparison

You can compare field values with specified numeric values by using the following operators: > , >= , = , < , and <= .

e_search('age	>= 18')	#	>=18
e_search('age	> 18')	#	> 18
e_search('age	= 18')	#	= 18
e_search('age	<= 18')	#	<=18
e search('age	< 18')	#	< 18

• Numeric range comparison

You can search for field values that are within a closed interval. An asterisk (*) can be used to specify an infinite interval.

Search for strings by judging logical relationships

Logical operators can be used among multiple search conditions. Parentheses () are used to nest search conditions.

Logical operator	Keyword
AND	and , AND , and $_{\&\&}$. The keywords are not case-sensitive.
OR	or and OR . The keywords are not case-sensitive.
NOT	not , NOT , and ! . The keywords are not case-sensitive.

Examples:

e_search('abc OR xyz') # The logical operator is not case-sensitive. e_search('abc and (xyz or zzz)') e_search('abc and not (xyz and not zzz)') e_search('abc 46 xyz') # and e_search('abc || xyz') # or e_search('abc || !xyz') # or not

Local operators can also be used to match substrings.

e_search('field: (abc OR xyz)') # The field value contains abc or xyz. e_search('field: (abc OR not xyz)') # The field value contains abc or does not contain xyz. e_search('field: (abc && !xyz)') # The field value contains abc and does not contain xyz.

Field check

You can use query strings to check fields.

- e_search('field: *') : checks whether a field exists.
- e_search('not field:*') : checks whether a field does not exist.
- e_search('not field:""') : checks whether a field does not exist.
- e_search('field: "?"') : checks whether a field exists and whether the field is not empty.
- e search('field==""') : checks whether a field exists and whether the field is empty.
- e_search('field~=".+"') : checks whether a field exists and whether the field is not empty.
- e_search('not field~=".+"') : checks whether a field does not exist or whether the field is empty.
- e_search('not field==""') : checks whether a field does not exist or whether the field is not empty.

4.5.8.8.3. Field extraction modes

This topic describes the values of the mode parameter in different functions. The mode parameter specifies a field extraction mode.

Related functions

The following table describes the functions that use the mode parameter and the default value that is used for the mode parameter in each function.

Category	Function	Default value of mode
Value assignment functions	e_set	overwrite
	e_regex	fill-auto
	e_json	fill-auto
	e_kv	fill-auto
Value extraction functions	e_csv, e_psv, and e_tsv	fill-auto
	e_kv_delimit	fill-auto
	e_anchor	overwrite
	e_syslogrfc	overwrite
Mapping and enrichment functions	e_dict_map	fill-auto
	e_table_map	fill-auto
	e_search_dict_map	overwrite
	e_search_table_map	fill-auto

Field extraction check and overwrite modes

The following table describes the values of the mode parameter.

Value	Description
fill	Sets a new field if the field does not exist or if the field already exists but the value of the field is an empty string.
fill-auto	Sets a new field if the new value is not an empty string and one of the following conditions is met: The field does not exist. The field already exists but the value of the field is an empty string.
add	Sets a new field if the field does not exist.
add-auto	Sets a new field if the new value is not an empty string and the field does not exist.
overwrite	Always sets a new field.
overwrite-auto	Sets a new field if the new value is not an empty string.

Result

The following table provides examples on how the functions work in different modes.

Transformation rule

Raw log

a: # An empty string b: 100

Transformation examples

Mode

> Document Version: 20240703

add	e_set("c", "123", mode='add')	a:# An empty string b: 100 c: 123
	e_set("c", "", mode='add')	a:# An empty string b: 100 c:
	e_set("a", "123", mode='add')	a:# An empty string b: 100
add-auto	<pre>e_set("c", "", mode='add-auto')</pre>	The c field is not added, and the raw log remains unchanged.
fill	e_set("c", "123", mode='fill')	a:# An empty string b: 100 c: 123
	e_set("c", "", mode='fill')	a:# An empty string b: 100 c:
	e_set("a", "123", mode='fill')	a: 123 b: 100
	<pre>e_set("b", "123", mode='fill')</pre>	The b field remains b: 100 .
fill-auto	<pre>e_set("c", "", mode='fill-auto')</pre>	The c field is not added, and the raw log remains unchanged.
overwrite	<pre>e_set("c", "123", mode='overwrite')</pre>	a:# An empty string b: 100 c: 123
	e_set("c", "", mode='overwrite')	a:# An empty string b: 100 c:
	<pre>e_set("b", "200", mode='overwrite')</pre>	a:# An empty string b: 200
	<pre>e_set("b", "", mode='overwrite')</pre>	a:# An empty string b:
overwrite-auto	<pre>e_set("b", "", mode='overwrite-auto')</pre>	The b field remains b: 100 .

Limits on field names for extraction

Functions such as e_json, e_kv, e_kv_delimit, and e_regex are supported.

 $\begin{array}{l} \label{eq:constraint} \text{Only the fields whose names abide by the limits can be extracted. The fields whose names do not abide by the limits are discarded. The regular expression <code>u'_*[\u4e00-\u9fa5\u0800-\u4e00a-zA-Z][\u4e00-\u9fa5\u0800-\u4e00\\w\\.\\-]*' is not supported. For example, the fields whose names match <code>123=abc _1_:100 lk=200 {"123": "456"} are discarded. \end{array}$ </code></code>

The following example shows how to use the default limits of a function on field names:

```
    Raw log
```

```
data: {"k1": 100, "k2": {"k3": 200, "k4": {"k5": 300} } }
• Transformation rule
```

```
e_json(
    "data",
    fmt="parent",
    sep="0",
    prefix="__",
    suffix="__",
    include_node=r"[\u4e00-\u9fa5\u0800-\u4e00a-zA-Z][\w\-\.]*",
    mode="fill-auto",
)
```

```
    Result
```

data: {"k1": 100, "k2": {"k3": 200, "k4": {"k5": 300} } }
data@_k1_:100
k2@_k3_:200
k4@_k5_:300

4.5.8.8.4. Regular expressions

This topic describes the matching modes of regular expressions and the methods that can be used to escape special characters in regular expressions.

Full match

If a regular expression matches an entire string, a full match is performed. For example, Vd+ fully matches 1234.

Some functions support partial matches for regular expressions. To perform full matches, you can enclose the regular expressions by using a caret ($^{\circ}$) and a dollar sign ($^{\circ}$) in the ^Regular expressions format.

The following table describes the matching modes for different functions.

Category	Function	Matching mode
	e_regex	Partial match
	e_keep_fields	Full match
	e_drop_fields	Full match
Global processing functions	e_rename	Full match
	e_kv	Partial match
	e_search_dict_map	Partial match
	e_search_dict_map	Partial match
Expression functions	e_match	Full match by default (configurable by using a parameter)
	e_search	Partial match
	regex_select	Partial match
	regex_findall	Partial match
	regex_match	Partial match by default (configurable by using a parameter)
	regex_replace	Partial match
	regex_split	Partial match

Examples:

- regex_match ("abc123", r"\d+") : The string matches the regular expression. In this example, the default matching mode of partial match is used.
- regex_match ("abc123", r"\d+", full=True) : The string does not match the regular expression. In this example, the matching mode is set to full match.
- regex_match ("abc123", r"^\d+\$") : The string does not match the regular expression. In this example, the matching mode is considered full match.
- e_search(r'status~="\d+"') : Whether the value of the **status** field matches the regular expression is based on the actual value. In this example, the matching mode is considered partial match.
- e_search(r'status=="^\d+\$"') : Whether the value of the status field matches the regular expression is based on the actual value. In this example, the matching mode is considered full match.

Character escape

Regular expressions may contain special characters. If you want to retain the literal meanings of the characters, you must escape the characters. You can use the following methods to escape special characters:

- Use backslashes (\).
- For more information, see Escape special characters.
- Use the str_regex_escape function.
- Example 1: If you use e_drop_fields(str_regex_escape("abc.test") , the **abc.test** field is discarded.
- Example 2: If you use e_drop_fields("abc.test"), the fields that match abc?test are discarded. The question mark (?) specifies any character.

Group

You can use parentheses () to enclose subexpressions in a regular expression to create a group. The group can be repeatedly referenced. The following example shows the difference between a regular expression before and after a group is created:

```
Log before processing:
SourceIP: 192.0.2.1
Log after processing:
SourceIP: 192.0.2.1
ip: 192.0.2.1
"""
# Before a group is created:
e_regex("SourceIP",r"\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}","ip")
# After a group is created:
e_regex("SourceIP", "\d{1,3}(.\d{1,3}){3}", "ip")
```

Capturing group

The text content that matches a capturing group is cached in the memory. The matched text content can be reused in other regular expressions by using backreferences. If the content that is enclosed in the parentheses

By default, all capturing groups are numbered from left to right based on an opening parenthesis. The first group is numbered 1, the second group is numbered 2, and so on. In the following example, three capturing groups are created:

 $(d{4}) - (d{2} - (d{2}))$ 1 1 2 3 32

If a regular expression contains both common capturing groups and named capturing groups, the named capturing groups are numbered after the common capturing groups. Log Service allows you to directly reference the custom name of a capturing group in regular expressions or programs.

Non-capturing group

The text content that matches a non-capturing group is not cached in the memory. If the content that is enclosed in the parentheses () of a group starts with 2: , the group is a non-capturing group.

For example, if you want to search for **program** and **project**, you can use the pro(gram|ject) regular expression. If you do not want to cache the content that matches the regular expression in the memory, you can use pro(?:gram|ject).

() Note (?:x) specifies that x matches the content but the matched content is not cached. You can define a subexpression in the (?:x) format and use the subexpression together with operators in the regular expression.

4.5.8.8.5. Grok patterns

Grok is a tool that combines multiple predefined regular expressions to match and split text and map the text segments to keys. Grok can be used to process log data. This topic describes the Grok patterns and provides several examples of basic syntax. The following table lists the Grok patterns.

() Note Named parameters cannot be used in some Grok patterns and their combinations, such as SYSLOGBASE, COMMONAPACHELOG, COMBINEDAPACHELOG, HTTPD20_ERRORLOG, HTTPD24_ERRORLOG, and HTTPD_ERRORLOG.

Туре	Pattern	Description
	EXTRACTJSON	Matches JSON data.
	CHINAID	Matches the numbers of identity cards of Chinese residents.
	USERNAME	Matches content that contains letters, digits, and $\ \ ,\$.
	USER	Matches content that contains letters, digits, and $\ \ ._^-$.
	EMAILLOCALPART	Matches the characters before the at sign ($\textcircled{0}$) in an email address. For example, in the email address username@example.com, the matched content is username.
	EMAILADDRESS	Matches email addresses.
	HTTPDUSER	Matches email addresses or usernames.
	INT	Matches integers.
Common patterns	BASE10NUM	Matches decimal numbers.
	NUMBER	Matches numbers.
	BASE16NUM	Matches hexadecimal numbers.
	BASE16FLOAT	Matches hexadecimal floating-point numbers.
	POSINT	Matches positive integers.
	NONNEGINT	Matches non-negative integers.
	WORD	Matches letters, digits, and underscores (_).
	NOTSPACE	Matches characters that are not spaces.
	SPACE	Matches spaces.
	DATA	Matches line feeds.
	GREEDYDATA	Matches zero or multiple characters that are not line feeds.
	QUOTEDSTRING	Matches quoted content. For example, in the ${\tt I}$ am "Iron Man" string, the matched content is ${\tt Iron\ Man}$.
	UUID	Matches universally unique identifiers (UUIDs).
	MAC	Matches MAC addresses.
	CISCOMAC	Matches Cisco MAC addresses.
	WINDOWSMAC	Matches Windows MAC addresses.
	COMMONMAC	Matches common MAC addresses.
	IPV6	Matches IPv6 addresses.
Networking	IPV4	Matches IPv4 addresses.

	IP	Matches IPv6 or IPv4 addresses.
	HOSTNAME	Matches hostnames.
	IPORHOST	Matches IP addresses or hostnames.
	HOSTPORT	Matches IP addresses, hostnames, or positive integers.
	PATH	Matches UNIX paths or Windows paths.
	UNIXPATH	Matches UNIX paths.
	WINPATH	Matches Windows paths.
		Matches IIRI schemes For example in http://bostname.domain.tld/_astats?
Paths	URIPROTO	application=&inf.name=eth0 , the matched content is http .
	ΤΤΥ	Matches tty paths.
	URIHOST	<pre>Matches IP addresses, hostnames, or positive integers. For example, in http://hostname.domain.tld/_astats?application=&inf.name=eth0 , the matched content is</pre>
	URI	Matches URIs.
	MONTH	Matches months that are in the numeric, abbreviated, or full-name format.
	MONTHNUM	Matches months that are in the numeric format.
Date	MONTHDAY	Matches days in a month.
	DAY	Matches weekdays that are in the abbreviated or full-name format.
	YEAR	Matches years.
	HOUR	Matches hours.
	MINUTE	Matches minutes.
	SECOND	Matches seconds.
	TIME	Matches time.
	DATE_US	Matches dates in the Month-Day-Year or Month/Day/Year format.
	DATE_EU	Matches dates in the Day-Month-Year, Day/Month/Year or Day.Month.Year format.
	ISO8601_TIMEZONE	Matches hours and minutes that are in the ISO 8601 format.
	ISO8601_SECOND	Matches seconds that are in the ISO 8601 format.
Time	TIMESTAMP_ISO8601	Matches time that is in the ISO 8601 format.
	DATE	Matches dates that are in the US or EU format.
	DATESTAMP	Matches dates and time.
	TZ	Matches UTC time zones.
	DATESTAMP_RFC822	Matches time that is in the RFC 822 format.
	DATESTAMP_RFC2822	Matches time that is in the RFC 2822 format.
	DATESTAMP_OTHER	Matches time that is in other formats.
	DATESTAMP_EVENTLOG	Matches time that is in the EventLog format.
	HTTPDERROR_DATE	Matches time that is in the httpd error format.
	SYSLOGTIMESTAMP	Matches time that is in the Syslog format.
SYSLOG	PROG	Matches programs.
	SYSLOGPROG	Matches programs and process identifiers (PIDs).
	SYSLOGHOST	Matches IP addresses or hostnames.
	SYSLOGFACILITY	Matches facilities.
	HTTPDATE	Matches dates and time that are in the HTTP format.
LOGFORMATL	LOGFORMAT	Matches Syslog logs that are in the traditional format.
	COMMONAPACHELOG	Matches common Apache logs.
	COMBINEDAPACHELOG	Matches combined Apache logs.
	HTTPD20_ERRORLOG	Matches httpd20 logs.
	HTTPD24_ERRORLOG	Matches httpd24 logs.
	HTTPD_ERRORLOG	Matches httpd logs.
LOGLEVELS	LOGLEVELS	Matches log levels, such as warn and debug.

General GROK patterns

EXTRACTJSON (?<json>(?:\{\s*"(?:\\"|[^"])+"\s*:\s*(?:(?P>json)|"(?:\\"|[^"])+"|[-+]?(0|[1-9]\d*)(?:\.[-+]?(0|[1-9]\d*))?(?:[eE][-+]?(0|[1-9]\d*))?(?:[eE][-+]?(0)[1-9]\d*))?(? 9]\d*))?(?:true|false)|null)(?:\s*,\s*"(?:\\"|[^"])+"\s*:\s*(?:(?P>json)|"(?:\\"|[^"])+"\[-+]?(0|[1-9]\d*))?(?:[eE][- $+]?(0|[1-9]\d^*))?(?:true|false)|null))*\s^*)|[(s^*(?:(?E>json)|"(?:\)"|[^*"])+"|[-+]?(0|[1-9]\d^*)(?:\[-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eE][-+]?(0|[1-9]\d^*))?(?:[eEE][-+]?(0|[1-9]\d^*))?(?:[eEE][-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?:[eEE[[-+]?(0|[1-9]\d^*))?(?$ $9 \) \) \) \ (?: true \ | false) \ | null) \ (?: \ (?: \ (?: \) \) \) \) \ " \ (?: \ (?: \) \ ()$ (?:true|false)|null))*\s*\])) CHINAID [1-9]\d{5})((18|19|([23]\d))\d{2}((0[1-9])(10|11|12))(([0-2][1-9])|10|20|30|31)\d{3}[0-9xx]\$)|(^[1-9]\d{5}\d{2}((0[1-9])(10|11|12))(([0-2][1-9])|10|20|30|31)\d{3} USERNAME [a-zA-Z0-9._-]+ USER %{USERNAME} EMAILLOCALPART [a-zA-Z][a-zA-Z0-9 .+-=:]+ EMAILADDRESS %{EMAILLOCALPART}@%{HOSTNAME} HTTPDUSER %{EMAILADDRESS}|%{USER} INT (?:[+-]?(?:[0-9]+)) BASE10NUM (?<![0-9.+-])(?>[+-]?(?:(?:[0-9]+(?:\.[0-9]+)?)|(?:\.[0-9]+))) NUMBER (?:%{BASE10NUM}) BASE16NUM (?<![0-9A-Fa-f])(?:[+-]?(?:0x)?(?:[0-9A-Fa-f]+)) BASE16FLOAT \b(?<![0-9A-Fa-f]))(?:[+-]?(?:0x)?(?:(?:[0-9A-Fa-f]+(?:\.[0-9A-Fa-f]*)?))(?:\.[0-9A-Fa-f]+))))b POSINT \b(?:[1-9][0-9]*)\b NONNEGINT \b(?:[0-9]+)\b WORD \b\w+\b NOTSPACE \S+ SPACE \s* DATA .*? GREEDYDATA .* OUOTEDSTRING (?>(?<!\\) (?>"(?>\\.|[^\\"]+)+"|""|(?>'(?>\\.|[^\\']+)+')|''|(?>`(?>\\.|[^\\']+)+`)|``)) UUID [A-Fa-f0-9] {8}-(?:[A-Fa-f0-9] {4}-) {3} [A-Fa-f0-9] {12} """Networking""" MAC (?:%{CISCOMAC}|%{WINDOWSMAC}|%{COMMONMAC}) CISCOMAC (?:(?:[A-Fa-f0-9]{4}\.){2}[A-Fa-f0-9]{4}) WINDOWSMAC (?:(?:[A-Fa-f0-9]{2}-){5}[A-Fa-f0-9]{2}) COMMONMAC (?:(?:[A-Fa-f0-9]{2}:){5}[A-Fa-f0-9]{2}) $a-f] \{1,4\}) \{0,4\}: ((25[0-5]|2[0-4] d|1 d d | [1-9]? d) ((.25[0-5]|2[0-4] d | 1 d d | [1-9]? d)) \{3\})) | :)) | (:((:[0-9A-Fa-f] \{1,4\}) \{1,7\}) | (:(:[0-9A-Fa-f] \{1,4\}) \{1,7\}) | (:(:[0-9A-Fa-f] \{1,4\}) \{1,7\}) | (::[0-9A-Fa-f] \{1,4\}) \{1,7\}) | (::[0-9A-Fa-f] \{1,4\}) \{1,7\}) | (::[0-9A-Fa-f] \{1,4\}) \{1,7\} | (::[0-9A-Fa-f] \{1,4\}) \{1,7\}) | (::[0-9A-Fa-f] \{1,4\}) \{1,7\} | (::[0-9A-Fa-f] \{1,4\}) | (::[0-9A-Fa-f] \{1,4\}) | (::[0-9A-Fa-f] \{1,4\}) | (::[0-9A-Fa-f] | (::[0$ $\{1,4\}$ $\{0,5\}$: ((25[0-5] 2[0-4] d 1 d 1 d [1-9]? d) (\. (25[0-5] 2[0-4] d 1 d [1-9]? d) (3)) :))) (%.+)? IPV4 (?<![0-9])(?:(?:[0-1]?[0-9]{1,2}]2[0-4][0-9]|25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9]|25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9]]25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0) [.] (?: [0-1]?[0-9] {1,2} | 2[0-4] [0-9] | 25[0-5])) (?! [0-9]) IP (?:%{IPV6}|%{IPV4}) HOSTNAME \b(?:[0-9A-Za-z][0-9A-Za-z-]{0,62})(?:\.(?:[0-9A-Za-z][0-9A-Za-z-]{0,62}))*(\.?|\b) IPORHOST (?:%{IP}|%{HOSTNAME}) HOSTPORT %{IPORHOST}:%{POSINT} """paths""" PATH (?:%{UNIXPATH} |%{WINPATH}) UNIXPATH (/([\w_%!\$@:.,~-]+|\\.)*)+ TTY (?:/dev/(pts|tty([pq])?)(\w+)?/?(?:[0-9]+)) WINPATH (?>[A-Za-z]+: |\\)(?:\\[^\\?*]*)+ URIPROTO [A-Za-z]+(\+[A-Za-z+]+)? URIHOST %{IPORHOST}(?::%{POSINT:port})? """uripath comes loosely from RFC1738, but mostly from what Firefox""" """doesn't turn into %XX""" URIPATH (?:/[A-Za-z0-9\$.+!*'(){},~:;=@#%_\-]*)+ URIPARAM \?[A-Za-z0-9\$.+!*'|(){},~@#%&/=:;_?\-\[\]<>]* URIPATHPARAM %{URIPATH}(?:%{URIPARAM})? URI %{URIPROTO};//(?:%{USER}(?::[^@]*)?@)?(?:%{URIHOST})?(?:%{URIPATHPARAM})? """ Months: January, Feb, 3, 03, 12, December""" MONTH \b(?:Jan(?:uary|uar)?|Feb(?:ruary|ruar)?|M(?:a|ä)?r(?:ch|z)?|Apr(?:il)?|Ma(?:y|i)?|Jun(?:e|i)?|Jul(?:y)?|Aug(?:ust)?|Sep(?:tember)? IO(?:c|k)?t(?:ober)?INov(?:ember)?IDe(?:c|z)(?:ember)?)\b MONTHNUM (?:0?[1-9]|1[0-2]) MONTHNUM2 (?:0[1-9]|1[0-2]) MONTHDAY (?:(?:0[1-9]) | (?:[12][0-9]) | (?:3[01]) | [1-9]) """Days: Monday, Tue, Thu, etc..."" DAY (?:Mon(?:day)?|Tue(?:sday)?|Wed(?:nesday)?|Thu(?:rsday)?|Fri(?:day)?|Sat(?:urday)?|Sun(?:day)?) """ Years?"" YEAR (?>\d\d) {1,2} HOUR (?:2[0123]|[01]?[0-9]) MINUTE (?:[0-5][0-9]) """'60' is a leap second in most time standards and thus is valid.""" SECOND (?:(?:[0-5]?[0-9]|60)(?:[:.,][0-9]+)?) TIME (?!<[0-9])%{HOUR}:%{MINUTE}(?::%{SECOND})(?![0-9]) """datestamp is YYYY/MM/DD-HH:MM:SS.UUUU (or something like it)""" $\texttt{DATE_US } {\texttt{MONTHNUM} [/-] } {\texttt{MONTHDAY} [/-] } {\texttt{YEAR}}$ DATE EU %{MONTHDAY}[./-]%{MONTHNUM}[./-]%{YEAR} IS08601 SECOND (?:%{SECOND}|60)

Cloud Defined Storage

TIMESTAMP ISO8601 %(YEAR)-%(MONTHNUM)-%(MONTHDAY)[T]%(HOUR):?%(MINUTE)(?::?%(SECOND))?%(ISO8601 TIMEZONE)? _ DATE %{DATE_US}|%{DATE_EU} DATESTAMP %{DATE}[-]%{TIME} TZ (?:[PMCE][SD]T|UTC) DATESTAMP RFC822 %{DAY} %{MONTH} %{MONTHDAY} %{YEAR} %{TIME} %{TZ} DATESTAMP RFC2822 %{DAY}, %{MONTHDAY} %{MONTH} %{YEAR} %{TIME} %{ISO8601 TIMEZONE} DATESTAMP_OTHER %{DAY} %{MONTH} %{MONTHDAY} %{TIME} %{TZ} %{YEAR} _ DATESTAMP EVENTLOG %{YEAR}%{MONTHNUM2}%{MONTHDAY}%{HOUR}%{MINUTE}%{SECOND} HTTPDERROR_DATE %{DAY} %{MONTH} %{MONTHDAY} %{TIME} %{YEAR} """Syslog Dates: Month Day HH:MM:SS"" SYSLOGTIMESTAMP %{MONTH} +%{MONTHDAY} %{TIME} PROG [\x21-\x5a\x5c\x5e-\x7e]+ SYSLOGPROG %{PROG:program}(?:\[%{POSINT:pid}\])? SYSLOGHOST %{IPORHOST} SYSLOGFACILITY <%{NONNEGINT:facility}.%{NONNEGINT:priority}> HTTPDATE %{MONTHDAY}/%{MONTH}/%{YEAR}:%{TIME} %{INT} """Shortcuts""" QS %{QUOTEDSTRING} """Log formats""" SYSLOGBASE %{SYSLOGTIMESTAMP:timestamp} (?:%{SYSLOGFACILITY})?%{SYSLOGHOST:logsource} %{SYSLOGPROG}: COMMONAPACHELOG %{IPORHOST:clientip} %{HTTPDUSER:ident} %{USER:auth} \[%{HTTPDATE:timestamp}\] "(?:%{WORD:verb} %{NOTSPACE:request}(?: HTTP/%{ NUMBER:httpversion})?|%{DATA:rawrequest})" %{NUMBER:response} (?:%{NUMBER:bytes}|-) COMBINEDAPACHELOG %{COMMONAPACHELOG} %{QS:referrer} %{QS:agent} HTTPD20_ERRORLOG \[%(HTTPDERROR_DATE:timestamp)\] \{%(LOGLEVEL:loglevel)\] (?:\[client %(IPORHOST:clientip)\])(0,1)%(GREEDYDATA:errormsg) HTTPD24_ERRORLOG \[%{HTTPDERROR_DATE:timestamp}\] \[%{WORD:module}:%{LOGLEVEL:loglevel}\] \[pid %{POSINT:pid}:tid %{NUMBER:tid}\](\(% {POSINT:proxy_errorcode}\)%{DATA:proxy_errormessage}:)?(\[client %{IPORHOST:client}:%{POSINT:clientport}\])? %{DATA:errorcode}: {GREEDYDATA:message} HTTPD ERRORLOG %{HTTPD20 ERRORLOG} |%{HTTPD24 ERRORLOG} "Log Levels"" LOGLEVEL ([Aa]lert|ALERT|[Tt]race|TRACE|[Dd]ebug|DEBUG|[Nn]otice|NOTICE|[Ii]nfo|INFO|[Ww]arn?(?:ing)?|WARN?(?:ING)?|[Ee]rr?(?:or)?|ERR?(?:OR)? | [Cc]rit?(?:ical)?|CRIT?(?:ICAL)?|[Ff]atal|FATAL|[Ss]evere|SEVERE|EMERG(?:ENCY)?|[Ee]merg(?:ency)?) Examples of basic syntax $([0-2][1-9])|10|20|30|31) d{3}$ USERNAME [a-zA-Z0-9. -]+ USER %{USERNAME} EMAILLOCALPART [a-zA-Z][a-zA-Z0-9_.+-=:]+ EMAILADDRESS %{EMAILLOCALPART}@%{HOSTNAME} HTTPDUSER %{EMAILADDRESS}|%{USER} INT (?:[+-]?(?:[0-9]+)) BASE10NUM (?<![0-9.+-])(?>[+-]?(?:(?:[0-9]+(?:\.[0-9]+)?)|(?:\.[0-9]+))) NUMBER (?:%{BASE10NUM}) BASE16NUM (?<![0-9A-Fa-f])(?:[+-]?(?:0x)?(?:[0-9A-Fa-f]+)) BASE16FLOAT \b(?<![0-9A-Fa-f])(?:[+-]?(?:0x)?(?:(?:[0-9A-Fa-f]+(?:\.[0-9A-Fa-f]*)?)|(?:\.[0-9A-Fa-f]+)))\b POSINT \b(?:[1-9][0-9]*)\b NONNEGINT \b(?:[0-9]+) \b WORD \b\w+\b NOTSPACE \S+ SPACE \s* DATA .*? GREEDYDATA $\texttt{QUOTEDSTRING} \ (2 > (2 < ! \) \ (2 > " (2 > \) \ | (^) ") + " | "" | (2 > ' (2 > \) \ | (^) + ") | ' | (2 > ` (2 > \) \ ((^) +) |))))$ UUID [A-Fa-f0-9] {8}-(?:[A-Fa-f0-9] {4}-) {3} [A-Fa-f0-9] {12} """Networking"" MAC (?:%{CISCOMAC}|%{WINDOWSMAC}|%{COMMONMAC}) CISCOMAC (?:(?:[A-Fa-f0-9]{4}\.){2}[A-Fa-f0-9]{4}) WINDOWSMAC (?:(?:[A-Fa-f0-9]{2}-){5}[A-Fa-f0-9]{2}) COMMONMAC (?:(?:[A-Fa-f0-9]{2}:){5}[A-Fa-f0-9]{2}) 5] 2[0-4] \d|1 \d|(1-9]?\d)) {3}) :)) | (([0-9A-Fa-f]{1,4}); {5} (((:[0-9A-Fa-f]{1,4}); {5}) ((:[25[0-5]] 2[0-4] \d|1 \d|1 \d|1 \d|[1-9]?\d) (\. (25[0-5]] 2[0-6] \d]2 [0-6] \d] $4 \left(1 - 9 \right) \left(3 \right) \left(1 - 9 \right) \left(3 \right) \left(1 - 9 \right) \right) \right) \left(1 - 9 \right) \right) \left(1 - 9 \right)$ (25[0-5] | 2[0-4] | d| | d| | 1-9] ? d)) | 3)) | .)) | (([0-9A-Fa-f] | 1, 4] .) | 3) (((:[0-9A-Fa-f] | 1, 4]) | (.(:[0-9A-Fa-f] | 1, 4]) | 0, 2) .) | (.(25[0-5] | 2[0-f] | 1, 4]) | 0, 2) .) | 0, 2) .:((25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)(\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)){3}))):))(([0-9A-Fa-f]{1,4}:){1}(((:[0-9A-Fa-f]{1,4})){1,6}))((:[0-9A-Fa-f]{1,6}))((:[0-9A-Fa-f]{1 $a-f] \{1,4\}) \{0,4\} : ((25[0-5]|2[0-4] d|1 d d [1-9]? d) ((.(25[0-5]|2[0-4] d |1 d d [1-9]? d)) \{3\})))))))) (:(((:[0-9A-Fa-f] \{1,4\}) \{1,7\})) ((:(0-9A-Fa-f) \{1,4\}) \{1,7\}) ((:(0-9A-Fa-f) \{1,4\}) \{1,7\})) ((:(0-9A-Fa-f) \{1,4\}) \{1,7\})) ((:(0-9A-Fa-f) \{1,4\}) \{1,7\})) ((:(0-9A-Fa-f) \{1,4\}) \{1,7\}) ((:(0-9A-Fa-f) \{1,4\}) ((1,7)) ((:(0-9A-Fa-f) \{1,4\}) ((1,7)) ((:(0-9A-Fa-f) \{1,4\}) ((1,7)) (((1,7)) ($ IPV4 (?<![0-9])(?:(?:[0-1]?[0-9]{1,2}]2[0-4][0-9]|25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9]|25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-9]{1,2}]2[0-4][0-9][25[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-1]?[0-5])[.](?:[0-

)[.](?:[0-1]?[0-9]{1,2}|2[0-4][0-9]|25[0-5]))(?![0-9]) IP (?:%{IPV6}|%{IPV4}))

HOSTNAME \b(?:[0-9A-Za-z][0-9A-Za-z-]{0,62})(?:\.(?:[0-9A-Za-z][0-9A-Za-z-]{0,62}))*(\.?|\b) HORHOST (?:%[IF]%(HOSTNAME))

HOSTPORT %{IPORHOST}:%{POSINT}

"""paths"""

PATH (?:%{UNIXPATH})%{WINPATH}) UNIXPATH (/ ([\w_§!\$e:,-]+\\.)*)+ TTY (?:/dev/(pts|tty([pq])?) (\w+)?/?(?:[0-9]+)) WINPATH (?>[A-Za-2]+:\\) (?:\\[^\\?*]*)+

User Guide-Log Service

URIPROTO [A-Za-z]+(+[A-Za-z+]+)?URIHOST %{IPORHOST}(?::%{POSINT:port})? """uripath comes loosely from RFC1738, but mostly from what Firefox""" """doesn't turn into %XX""" URIPATH (?:/[A-Za-z0-9\$.+!*'(){},~:;=@#%_\-]*)+ """URIPARAM \?(?:[A-Za-z0-9]+(?:=(?:[^&]*))?(?:&(?:[A-Za-z0-9]+(?:=(?:[^&]*))?)*)?""" URIPARAM \?[A-Za-z0-9\$.+!*'|(){},~@#%&/=:;_?\-\[\]<>]* URIPATHPARAM %{URIPATH}(?:%{URIPARAM})? URI %{URIPROTO}://(?:%{USER}(?::[^@]*)?@)?(?:%{URIHOST})?(?:%{URIPATHPARAM})? """ Months: January, Feb, 3, 03, 12, December""" MONTH \b(?:Jan(?:uary|uar)?|Feb(?:ruary|ruar)?|M(?:a|ä)?r(?:ch|z)?|Apr(?:il)?|Ma(?:y|i)?|Jun(?:e|i)?|Jul(?:y)?|Aug(?:ust)?|Sep(?:tember)? |O(?:c|k)?t(?:ober)?|Nov(?:ember)?|De(?:c|z)(?:ember)?)\b MONTHNUM (?:0?[1-9]|1[0-2]) MONTHNUM2 (?:0[1-9]|1[0-2]) MONTHDAY (?:(?:0[1-9]) | (?:[12][0-9]) | (?:3[01]) | [1-9]) """Days: Monday, Tue, Thu, etc...""" DAY (?:Mon(?:day)?|Tue(?:sday)?|Wed(?:nesday)?|Thu(?:rsday)?|Fri(?:day)?|Sat(?:urday)?|Sun(?:day)?) """ Years?""" YEAR (?>\d\d){1,2} HOUR (?:2[0123]|[01]?[0-9]) MINUTE (?:[0-5][0-9]) """'60' is a leap second in most time standards and thus is valid.""" SECOND (?:(?:[0-5]?[0-9]|60)(?:[:.,][0-9]+)?) TIME (?!<[0-9])%{HOUR}:%{MINUTE}(?::%{SECOND})(?![0-9]) """datestamp is YYYY/MM/DD-HH:MM:SS.UUUUU (or something like it)""" DATE US %{MONTHNUM}[/-]%{MONTHDAY}[/-]%{YEAR} DATE EU %{MONTHDAY}[./-]%{MONTHNUM}[./-]%{YEAR} ISO8601_TIMEZONE (?:Z|[+-]%{HOUR}(?::?%{MINUTE})) ISO8601_SECOND (?:%{SECOND}|60) $\texttt{TIMESTAMP_ISO8601 } {\texttt{VEAR}-\texttt{MONTHNUM}-\texttt{MONTHDAY}[T]} {\texttt{HOUR}:?\texttt{MINUTE}(?::?\texttt{SECOND})?\texttt{ISO8601_TIMEZONE}?$ DATE %{DATE_US}|%{DATE_EU} DATESTAMP %{DATE}[-]%{TIME} TZ (?:[PMCE][SD]T|UTC) DATESTAMP_RFC822 %{DAY} %{MONTH} %{MONTHDAY} %{YEAR} %{TIME} %{TZ} DATESTAMP_RFC2822 %{DAY}, %{MONTHDAY} %{MONTH} %{YEAR} %{TIME} %{ISO8601_TIMEZONE} DATESTAMP OTHER %{DAY} %{MONTH} %{MONTHDAY} %{TIME} %{TZ} %{YEAR} DATESTAMP EVENTLOG %{YEAR}%{MONTHNUM2}%{MONTHDAY}%{HOUR}%{MINUTE}%{SECOND} """Syslog Dates: Month Day HH:MM:SS""" $\label{eq:syslogTimestamp & (MONTH} + & (MONTHDAY) & (TIME) \\ \mbox{PROG } [\x21-\x5a\x5c\x5e-\x7e]+ \\ \end{tabular}$ SYSLOGPROG %{PROG:program}(?:\[%{POSINT:pid}\])? SYSLOGHOST %{IPORHOST} SYSLOGFACILITY <% {NONNEGINT: facility}.% {NONNEGINT: priority}> HTTPDATE %{MONTHDAY}/%{MONTH}/%{YEAR}:%{TIME} %{INT} """Shortcuts""" QS %{QUOTEDSTRING} """Log formats""" SYSLOGBASE %{SYSLOGTIMESTAMP:timestamp} (?:%{SYSLOGFACILITY}) ?%{SYSLOGHOST:logsource} %{SYSLOGPROG}: COMMONAPACHELOG %{IPORHOST:clientip} %{HTTPDUSER:ident} %{USER:auth} \[%{HTTPDATE:timestamp}\] "(?:%{WORD:verb} %{NOTSPACE:request}(?: HTTP/%{ NUMBER:httpversion})?|%{DATA:rawrequest})" %{NUMBER:response} (?:%{NUMBER:bytes}|-) COMBINEDAPACHELOG %{COMMONAPACHELOG} %{QS:referrer} %{QS:agent} HTTPD20_ERRORLOG \[%{HTTPDERROR_DATE:timestamp}\] \[%{LOGLEVEL:loglevel}\] (?:\[client %{IPORHOST:clientip}\]){0,1}%{GREEDYDATA:errormsg} HTTPD24_ERRORLOG \[%{HTTPDERROR_DATE:timestamp}\] \[%{WORD:module}:%{LOGLEVEL:loglevel}\] \[pid %{POSINT:pid}:tid %{NUMBER:tid}\](\(% {POSINT:proxy_errorcode}\)%{DATA:proxy_errormessage}:)?(\[client %{IPORHOST:client}:%{POSINT:clientport}\])? %{DATA:errorcode}: % {GREEDYDATA:message} HTTPD_ERRORLOG %{HTTPD20_ERRORLOG} | %{HTTPD24_ERRORLOG} """Log Levels""

LOGLEVEL ([Aa]lert|ALERT|[Tt]race|TRACE|[Dd]ebug|DEBUG|[Nn]otice|NOTICE|[Ii]nfo|INFO|[Ww]arn?(?:ing)?|WARN?(?:ING)?|[Ee]rr?(?:or)?|ERR?(?:OR)? | [Cc]rit?(?:ical)?|CRIT?(?:ICAL)?|[Ff]atal|FATAL|[Ss]evere|SEVERE|EMERG(?:ENCY)?|[Ee]merg(?:ency)?)

Sample patterns for AWS

S3_REQUEST_LINE (?:%{WORD:verb} %{NOTSPACE:request}(?: HTTP/%{NUMBER:httpversion})?|%{DATA:rawrequest})

S3_ACCESS_LOG %{WORD:owner} %{NOTSPACE:bucket} \{%{HTTPDATE:timestamp}\] %{IP:clientip} %{NOTSPACE:requester} %{NOTSPACE:request_id} %{NOTSPACE E:operation} %{NOTSPACE:key} (?:"%{S3_REQUEST_LINE}"|-) (?:%{INT:response:int}|-) (?:%{NOTSPACE:reror_code}) (?:%{INT:bytes:int}|-) (?:% {INT:object_size:int}|-) (?:%{INT:request_time_ms:int}|-) (?:%{INT:turnaround_time_ms:int}|-) (?:%{QS:referrer}|-) (?:"?%{QS:agent}"?|-) (?:-|%{NOTSPACE:version_id})

ELB_URIPATHPARAM %{URIPATH:path}(?:%{URIPARAM:params})?

ELB_URI %{URIPROTO:proto}://(?:%{USER}(?::[^@]*)?@)?(?:%{URIHOST:urihost})?(?:%{ELB_URIPATHPARAM})?

ELB REQUEST LINE (?:%{WORD:verb} %{ELB URI:request}(?: HTTP/%{NUMBER:httpversion})?|%{DATA:rawrequest})

ELB_ACCESS_LOG %{TIMESTAMP_IS08601:timestamp} %{NOTSPACE:elb} %{IP:clientip}:%{INT:clientport:int} (?:(%{IP:backendip}:?:%
{INT:backendport:int})|-) %{NUMBER:request_processing_time:float} %{NUMBER:backend_processing_time:float} %
{NUMBER:response_processing_time:float} %{INT:response:int} %{INT:backend_response:int} %{INT:received_bytes:int} %{INT:bytes:int} "%
{ELB_REQUEST_LINE}"

Sample patterns for Bacula

BACULA TIMESTAMP %{MONTHDAY}-%{MONTH} %{HOUR}:%{MINUTE} BACULA_HOST [a-zA-Z0-9-]+ BACULA_VOLUME %{USER} BACULA_DEVICE %{USER} BACULA DEVICEPATH %{UNIXPATH} BACULA CAPACITY %{INT}{1,3}(, %{INT}{3})* BACULA VERSION %{USER} BACULA JOB %{USER} BACULA_LOG_MAX_CAPACITY User defined maximum volume capacity %{BACULA_CAPACITY} exceeded on device \"%{BACULA_DEVICE;device}\" \(% {BACULA DEVICEPATH}\) BACULA LOG END VOLUME End of medium on Volume \"%{BACULA VOLUME:volume}\" Bytes=%{BACULA CAPACITY} Blocks=%{BACULA CAPACITY} at %{MONTHDAY}-%{ $\texttt{MONTH} = \{ \texttt{YEAR} \} \quad \{ \texttt{HOUR} \} : \{ \texttt{MINUTE} \} .$ BACULA_LOG_NEW_LABEL Labeled new Volume \"%{BACULA_VOLUME:volume}\" on device \"%{BACULA_DEVICE:device}\" \(%{BACULA_DEVICEPATH})). BACULA_LOG_WROTE_LABEL Wrote label to prelabeled Volume \"%{BACULA_VOLUME:volume}\" on device \"%{BACULA_DEVICE}\" \(%{BACULA_DEVICEPATH})) BACULA LOG NEW MOUNT New volume \"%{BACULA VOLUME:volume}\" mounted on device \"%{BACULA DEVICE:device}\" \(%{BACULA DEVICEPATH})) at % {MONTHDAY}-%{MONTH}-%{YEAR} %{HOUR}:%{MINUTE}. BACULA_LOG_NOOPEN \s+Cannot open %{DATA}: ERR=%{GREEDYDATA:berror} BACULA_LOG_NOOPENDIR \s+Could not open directory %{DATA}: ERR=%{GREEDYDATA:berror} BACULA LOG NOSTAT \s+Could not stat %{DATA}: ERR=%{GREEDYDATA:berror} BACULA LOG NOJOBS There are no more Jobs associated with Volume \"%{BACULA VOLUME:volume}\". Marking it purged. BACULA_LOG_ALL_RECORDS_PRUNED All records pruned from Volume \"%{BACULA_VOLUME:volume}\"; marking it \"Purged\" $\texttt{BACULA_LOG_BEGIN_PRUNE_JOBS}$ Begin pruning Jobs older than $IIT \ month \ INT \ days$. BACULA_LOG_BEGIN_PRUNE_FILES Begin pruning Files. BACULA_LOG_PRUNED_JOBS Pruned %{INT} Jobs* for client %{BACULA_HOST:client} from catalog. BACULA_LOG_PRUNED_FILES Pruned Files from %{INT} Jobs* for client %{BACULA HOST:client} from catalog. BACULA LOG ENDPRUNE End auto prune. BACULA_LOG_STARTJOB Start Backup JobId %{INT}, Job=%{BACULA_JOB:job} BACULA_LOG_STARTRESTORE Start Restore Job %{BACULA_JOB:job} BACULA_LOG_USEDEVICE Using Device \"%{BACULA_DEVICE:device}\" BACULA LOG DIFF FS \s+%{UNIXPATH} is a different filesvstem. Will not descend from %{UNIXPATH} into it. BACULA LOG JOBEND Job write elapsed time = %{DATA:elapsed}, Transfer rate = %{NUMBER} (K|M|G)? Bytes/second BACULA_LOG_NOPRUNE_JOBS No Jobs found to prune. BACULA LOG NOPRUNE FILES No Files found to prune. BACULA LOG READYAPPEND Ready to append to end of Volume \"%{BACULA VOLUME:volume}\" size=%{INT} BACULA LOG CANCELLING Cancelling duplicate JobId=%{INT}. BACULA_LOG_MARKCANCEL JobId %{INT}, Job %{BACULA_JOB:job} marked to be canceled. BACULA LOG CLIENT RBJ shell command: run ClientRunBeforeJob \"%{GREEDYDATA:runjob}\" BACULA_LOG_VSS (Generate)?VSS (Writer)? BACULA LOG MAXSTART Fatal error: Job canceled because max start delay time exceeded. BACULA LOG DUPLICATE Fatal error: JobId %{INT:duplicate} already running. Duplicate job not allowed. BACULA_LOG_NOJOBSTAT Fatal error: No Job status returned from FD. BACULA_LOG_FATAL_CONN Fatal error: bsock.c:133 Unable to connect to (Client: %{BACULA_HOST:client}|Storage daemon) on %{HOSTNAME}:%{POSINT}. E RR=(?<berror>%{GREEDYDATA}) BACULA LOG NO CONNECT Warning: bsock.c:127 Could not connect to (Client: %{BACULA HOST:client}|Storage daemon) on %{HOSTNAME}:%{POSINT}. ERR=(?<berror>%{GREEDYDATA}) BACULA_LOG_NO_AUTH Fatal error: Unable to authenticate with File daemon at %{HOSTNAME}. Possible causes: BACULA_LOG_NOSUIT No prior or suitable Full backup found in catalog. Doing FULL backup. BACULA_LOG_NOPRIOR No prior Full backup Job record found. BACULA LOG JOB (Error:)?Bacula % BACULA HOST } % BACULA VERSION \ (% BACULA VERSION \): BACULA_LOGLINE %{BACULA_INESTAMP:bts} %{BACULA_HOST:hostname} Jobid %{INT:jobid}: (%{BACULA_LOG_MAX_CAPACITY}){%{BACULA_LOG_END_VOLUME}}%{BACULA_LOG_END_VOLUME}}

(BACULA_LOG_NOISHI) *{BACULA_LOG_NOODS) *{BACULA_LOG_ALL_RECORDS **{BACULA_LOG_BECIN_PROME_JODS) **{BACULA_LOG_BECIN_PROME_FILES} ** (BACULA_LOG_REUNED_JOBS) **{BACULA_LOG_PRUNED_FILES} **{BACULA_LOG_ENDPRUNE} ** (BACULA_LOG_USEDEVICE) **{BACULA_LOG_DIFF_FS} ** (BACULA_LOG_USEDEVICE) **{BACULA_LOG_DIFF_FS} ** (BACULA_LOG_VOLUME_REVWRITTEN) **{BACULA_LOG_REDVAPPEND} ** (BACULA_LOG_CACELLINE) **{BACULA_LOG_MAXSTART} ** (BACULA_LOG_VOSS) **{BACULA_LOG_MAXSTART} ** (BACULA_LOG_DUPLICATE) **{BACULA_LOG_DUPLICATE} ** (BACULA_LOG_NO_AUTH) ** (BACULA_LOG_NO_AUTH) ** (BACULA_LOG_NOSUIT) ** (BACULA_LOG_NOSUIT) ** (BACULA_LOG_NOFICR))

Sample patterns for Bro
"""https://www.bro.org/sphinx/script-reference/log-files.html"""

"""http.log"""

 $BR0_HTTP \\ {NUMBER:ts} \\ t \\ {NTSPACE:uid} \\ t \\ {IP:orig_p} \\ t \\ {INT:orig_p} \\ t \\ {INT:resp_h} \\ t \\ {INT:resp_p} \\ t \\ {INT:resp_d} \\ t \\ {INT:resp_p} \\ t \\ {$ GREEDYDATA:domain}\t%{GREEDYDATA:uri}\t%{GREEDYDATA:referrer}\t%{GREEDYDATA:user_agent}\t%{NUMBER:request_body_len}\t* {NUMBER:response_body_len}\t%{GREEDYDATA:status_code}\t%{GREEDYDATA:status_msg}\t%{GREEDYDATA:info_code}\t%{GREEDYDATA:info_msg}\t% {GREEDYDATA:filename}\t%{GREEDYDATA:bro tags}\t%{GREEDYDATA:username}\t%{GREEDYDATA:password}\t%{GREEDYDATA:proxied}\t% {GREEDYDATA:orig fuids}\t%{GREEDYDATA:orig mime types}\t%{GREEDYDATA:resp fuids}\t%{GREEDYDATA:resp mime types}

"""dns.log"""

 $\label{eq:BR0_DNS ${NUMBER:ts}t} (NOTSPACE:uid) t {IP:orig_h} t {INT:orig_p} t {INT:resp_h} t {NORD:proto} t {INT:trans_id} t {INT:resp_h} t {INT:resp_h}$ {GREEDYDATA:query}\t%{GREEDYDATA:qclass}\t%{GREEDYDATA:qclass name}\t%{GREEDYDATA:qtype}\t%{GREEDYDATA:qtype name}\t%{GREEDYDATA:rcode}\t% {GREEDYDATA:rcode name}\t%{GREEDYDATA:A}\t%{GREEDYDATA:TC}\t%{GREEDYDATA:RD}\t%{GREEDYDATA:R}\t%{GREEDYDATA:Z}\t%{GREEDYDATA:A} {GREEDYDATA:TTLs}\t%{GREEDYDATA:rejected}

"""conn.log"""

 $BR0_CONN $ (NUMBER:ts) t (NOTSPACE:uid) t (IP:orig_h) t (INT:orig_p) t (IP:resp_h) t (INT:resp_p) t (INT:resp_h) t (INT:resp$ {NUMBER:duration}\t%{NUMBER:orig bytes}\t%{NUMBER:resp bytes}\t%{GREEDYDATA:conn state}\t%{GREEDYDATA:local orig}\t% {GREEDYDATA:missed_bytes}\t%{GREEDYDATA:history}\t%{GREEDYDATA:orig_pts}\t%{GREEDYDATA:orig_ip_bytes}\t%{GREEDYDATA:resp_pts}\t% {GREEDYDATA:resp ip bytes}\t%{GREEDYDATA:tunnel parents}

"""files.log"""

BRO FILES %{NUMBER:ts}\t%{NOTSPACE:fuid}\t%{IP:tx hosts}\t%{IP:rx hosts}\t%{NOTSPACE:conn uids}\t%{GREEDYDATA:source}\t%{GREEDYDATA:depth}\t% GREEDYDATA:analyzers}\t%{GREEDYDATA:mime_type}\t%{GREEDYDATA:filename}\t%{GREEDYDATA:duration}\t%{GREEDYDATA:local_orig}\t% {GREEDYDATA:is_orig}\t%{GREEDYDATA:seen_bytes}\t%{GREEDYDATA:total_bytes}\t%{GREEDYDATA:seen_bytes}\t%

Sample patterns for Exim

EXIM MSGID [0-9A-Za-z] {6}-[0-9A-Za-z] {6}-[0-9A-Za-z] {2} EXIM FLAGS (<=|[-=>*]>|[*]{2}|==) EXIM_DATE %{YEAR:exim_year}-%{MONTHNUM:exim_month}-%{MONTHDAY:exim_day} %{TIME:exim time} EXIM PID \[%{POSINT}\] EXIM_EXCLUDE_TERMS (Message is frozen (Start|End) queue run | Warning: | retry time not reached | no (IP address host name) found for (IP address|host) | unexpected disconnection while reading SMTP command | no immediate delivery: |another process is handling this message) EXIM_REMOTE_HOST (H=(%{NOTSPACE:remote_hostname})?(\(%{NOTSPACE:remote_heloname}\))?\[%{IP:remote_host}\]) EXIM_INTERFACE (I=\[%{IP:exim_interface}\](:%{NUMBER:exim_interface_port})) EXIM_PROTOCOL (P=%{NOTSPACE:protocol}) EXIM_MSG_SIZE (S=%{NUMBER:exim_msg_size}) EXIM_HEADER_ID (id=%{NOTSPACE:exim_header_id}) EXIM SUBJECT (T=%{QS:exim subject})

Sample patterns for Cisco firewalls

""" NetScreen firewall logs"" NETSCREENSESSIONLOG %{SYSLOGTIMESTAMP:date} %{IPORHOST:device} %{IPORHOST}: NetScreen device_id=%{WORD:device_id}%{DATA}: start_time=% {QUOTEDSTRING:start_time} duration=%{INT:duration) policy_id=%{INT:policy_id} service=%{DATA:service} proto=%{INT:proto} src zone=% {WORD:src_zone} dst zone=%{WORD:dst_zone} action=%{WORD:action} sent=%{INT:sent} rcvd=%{INT:rcvd} src=%{IPORHOST:src_ip} dst=% {IPORHOST:dst_ip} src_port=%{INT:src_port} dst_port=%{INT:dst_port} src-xlated ip=%{IPORHOST:src_xlated_ip} port=%{INT:src_xlated_port} dst-xla ted ip=%{IPORHOST:dst xlated ip} port=%{INT:dst xlated port} session id=%{INT:session id} reason=%{GREEDYDATA:reason} """== Cisco ASA ==""" CISCO_TAGGED_SYSLOG ^<%{POSINT:syslog_pri}>%{CISCOTIMESTAMP:timestamp}(%{SYSLOGHOST:sysloghost})? ?: %%{CISCOTAG:ciscotag}: CISCOTIMESTAMP %{MONTH} +%{MONTHDAY}(?: %{YEAR})? %{TIME} CISCOTAG [A-Z0-9]+-%{INT}-(?:[A-Z0-9]+) """Common Particles""" CISCO_ACTION Built|Teardown|Deny|Denied|denied|requested|permitted|denied by ACL|discarded|est-allowed|Dropping|created|deleted CISCO_REASON Duplicate TCP SYN|Failed to locate egress interface|Invalid transport field|No matching connection|DNS Response|DNS Query| (?:%{WO RD}\s*)* CISCO DIRECTION Inbound | inbound | Outbound | outbound CISCO INTERVAL first hit |% [INT}-second interval CISCO_XLATE_TYPE static|dynamic """ASA-1-104001""" CISCOFW104001 \(((?:Primary|Secondary))) Switching to ACTIVE - %{GREEDYDATA:switch reason} """ASA-1-104002"" CISCOFW104002 \((?:Primary|Secondary))) Switching to STANDBY - %{GREEDYDATA:switch reason} """ASA-1-104003"" CISCOFW104003 \((?:Primary|Secondary)\) Switching to FAILED\. """ASA-1-104004"" CISCOFW104004 \((?:Primary|Secondary)\) Switching to OK\. """ASA-1-105003"" CISCOFW105003 \((?:Primary|Secondary)\) Monitoring on [Ii]nterface %{GREEDYDATA:interface_name} waiting """ASA-1-105004""" CISCOFW105004 \(((?:Primary|Secondary)\) Monitoring on [Ii]nterface %{GREEDYDATA:interface_name} normal """ASA-1-105005"" CISCOFW105005 \(((?:Primary|Secondary)\) Lost Failover communications with mate on [Ii]nterface %{GREEDYDATA:interface name} """ASA-1-105008" CISCOFW105008 \((?:Primary|Secondary)\) Testing [Ii]nterface %{GREEDYDATA:interface_name} """ASA-1-105009"" CISCOFW105009 \(((?:Primary|Secondary)\) Testing on [Ii]nterface %{GREEDYDATA:interface name} (?:Passed|Failed) """ASA-2-106001"" CISCOFW106001 %{CISCO DIRECTION:direction} %{WORD; protocol} connection %{CISCO ACTION:action} from %{IP:src ip}/%{INT:src port} to % {IP:dst_ip}/%{INT:dst_port} flags %{GREEDYDATA:tcp_flags} on interface %{GREEDYDATA:interface} """ASA-2-106006, ASA-2-106007, ASA-2-106010"""

CISCOFW106006_106007_106010 %{CISCO_ACTION:action} %{CISCO_DIRECTION:direction} %{WORD:protocol} (?:from|src) %{IP:src_ip}/%{INT:src_port}(\(% {DATA:src_fwuser}\))? (?:to|dst) %{IP:dst_ip}/%{INT:dst_port}(\(%{DATA:dst_fwuser}\))? (?:on interface %{DATA:interface}|due to % {CISCO_REASON:reason}) """ASA-3-106014""" CISCOFW106014 %{CISCO ACTION:action} %{CISCO DIRECTION:direction} %{WORD:protocol} src %{DATA:src interface}:%{IP:src ip}(\{% {DATA:src_fwuser}\))? dst %{DATA:dst_interface}:%{IP:dst_ip}(\(%{DATA:dst_fwuser}))? \(type %{INT:icmp_type}, code %{INT:icmp_code})) """ASA-6-106015""" CISCOFW106015 %{CISCO_ACTION:action} %{WORD:protocol} \(%{DATA:policy_id}\) from %{IP:src_ip}/%{INT:src_port} to %{IP:dst_ip}/%{INT:dst_port} flags %{DATA:tcp_flags} on interface %{GREEDYDATA:interface} """ASA-1-106021"" CISCOFW106021 %{CISCO ACTION:action} %{WORD:protocol} reverse path check from %{IP:src ip} to %{IP:dst ip} on interface % {GREEDYDATA:interface} """ASA-4-106023""" CISCOFW106023 %{CISCO_ACTION:action}(protocol)? %{WORD:protocol} src %{DATA:src_interface}:%{DATA:src_ip}(/%{INT:src_port})?(\% {DATA:src_fwuser}\))? dst %{DATA:dst_interface}:%{DATA:dst_ip}//%{INT:dst_port})?(\(%{DATA:dst_fwuser})))?(\(type %{INT:icmp_type}, code %{INT}))? T:icmp_code}\))? by access-group "?%{DATA:policy_id}"? \[%{DATA:hashcode1}, %{DATA:hashcode2}]] """ASA-4-106100, ASA-4-106102, ASA-4-106103""" CISCOFW106100_2_3 access-list %{NOTSPACE:policy_id} %{CISCO_ACTION:action} %{WORD:protocol} for user '%{DATA:src_fwuser}' % {DATA:src_interface}/%{IP:src_ip}\(%{INT:src_port}\) -> %{DATA:dst_interface}/%{IP:dst_ip}\(%{INT:dst_port}\) hit-cnt %{INT:hit_count} %{CISCO _INTERVAL:interval} \[%{DATA:hashcode1}, %{DATA:hashcode2}\]
"""ASA-5-106100""" CISCOFW106100 access-list %{NOTSPACE:policy_id} %{CISCO_ACTION:action} %{MORD:protocol} %{DATA:src_interface}/%{IP:src_ip}\(%{INT:src_port}\)(\(%{DATA:src_fwuser}\))? -> %{DATA:dst_interface}/%{IP:dst_ip}\(%{INT:dst_port}\)(\(%{DATA:src_fwuser}\))? hit-cnt %{INT:hit_count} % {CISCO INTERVAL:interval} \[%{DATA:hashcode1}, %{DATA:hashcode2}\] """ASA-6-110002""" CISCOFW110002 %{CISCO REASON:reason} for %{WORD:protocol} from %{DATA:src interface}:%{IP:src ip}/%{INT:src port} to %{IP:dst ip}/% {INT:dst port} """ASA-6-302010""" CISCOFW302010 %{INT:connection_count} in use, %{INT:connection_count_max} most used """ASA-6-302013, ASA-6-302014, ASA-6-302015, ASA-6-302016""" CISCOFW302013_302014_302015_302016 &{CISCO_ACTION:action}(?: &{CISCO_DIRECTION:direction})? &{WORD:protocol} connection &{INT:connection_id} for %{DATA:src_interface}:%{IP:src_ip}/%{INT:src_port}(\(%{IP:src_mapped_ip}/%{INT:src_mapped_port}\))?(\(%{DATA:src_fwuser}\))? to % {DATA:dst_interface}:%{IP:dst_ip}/%{INT:dst_port}(\(%{IP:dst_mapped_ip}/%{INT:dst_mapped_ip}))?(\(%{DATA:dst_fwuser})))?(duration % {TIME:duration} bytes %{INT:bytes})?(?: %{CISCO_REASON:reason})?(\(%{DATA:user}\))? """ASA-6-302020, ASA-6-302021""" CISCOFW302020 302021 %{CISCO ACTION:action}(?: %{CISCO DIRECTION:direction})? %{WORD:protocol} connection for faddr %{IP:dst ip}/% {INT:icmp_seq_num}(?:\(%{DATA:fwuser})))? gaddr %{IP:src_xlated_ip}/%{INT:icmp_code_xlated} laddr %{IP:src_ip}/%{INT:icmp_code}(\ (% {DATA:user}\))? """ASA-6-305011""" CISCOFW305011 %{CISCO_ACTION:action} %{CISCO_XLATE_TYPE:xlate_type} %{WORD:protocol} translation from %{DATA:src_interface}:%{IP:src_ip}(/%{IN T:src port})?(\(%{DATA:src fwuser}\))? to %{DATA:src xlated interface}:%{IP:src xlated ip}/%{DATA:src xlated port} """ASA-3-313001, ASA-3-313004, ASA-3-313008"" CISCOFW313001_313004_313008 %{CISCO_ACTION:action} %{WORD:protocol} type=%{INT:icmp_type}, code=%{INT:icmp_code} from %{IP:src_ip} on interface %{DATA:interface}(to %{IP:dst_ip})? """ASA-4-313005""" CISCOFW313005 %{CISCO REASON:reason} for %{WORD:protocol} error message: %{WORD:err protocol} src %{DATA:err src interface}:%{IP:err src ip}(\ (%{DATA:err_src_fwuser}\))? dst %{DATA:err_dst_interface}:%{IP:err_dst_ip}(\(%{DATA:err_dst_fwuser}))? \(type %{INT:err_imp_type}, code %{IN T:err_icmp_code}\) on %{DATA:interface} interface\. Original IP payload: %{WORD:protocol} src %{IP:orig_src_ip}/%{INT:orig_src_port}(\(%{DATA :orig_src_fwuser}\))? dst %{IP:orig_dst_ip}/%{INT:orig_dst_port}(\(%{DATA:orig_dst_fwuser}\))? """ASA-5-321001""" CISCOFW321001 Resource '%{WORD:resource name}' limit of %{POSINT:resource limit} reached for system """ASA-4-402117"" CISCOFW402117 %{WORD:protocol}: Received a non-IPSec packet \(protocol= %{WORD:orig_protocol}\) from %{IP:src_ip} to %{IP:dst_ip} """ASA-4-402119""" CISCOFW402119 %{WORD:protocol}: Received an %{WORD:orig_protocol} packet \(SPI= %{DATA:spi}, sequence number= %{DATA:seq_num}\) from % {IP:src ip} \(user= %{DATA:user})) to %{IP:dst ip} that failed anti-replay checking """ASA-4-419001""" CISCOFW419001 %{CISCO_ACTION:action} %{WORD:protocol} packet from %{DATA:src_interface}:%{IP:src_ip}/%{INT:src_port} to % {DATA:dst_interface}:%{IP:dst_ip}/%{INT:dst_port}, reason: %{GREEDYDATA:reason} """ASA-4-419002""" CISCOFW419002 %{CISCO REASON:reason} from %{DATA:src interface}:%{IP:src ip}/%{INT:src port} to %{DATA:dst interface}:%{IP:dst ip}/% {INT:dst port} with different initial sequence number """ASA-4-500004""" CISCOFW500004 %{CISCO REASON:reason} for protocol=%{WORD:protocol}, from %{IP:src ip}/%{INT:src port} to %{IP:dst ip}/%{INT:dst port} """ASA-6-602303, ASA-6-602304""" CISCOFW602303_602304 %{WORD:protocol}: An %{CISCO_DIRECTION:direction} %{GREEDYDATA:tunnel_type} SA \(SPI= %{DATA:spi}\) between %{IP:src_ip} and %{IP:dst ip} \(user= %{DATA:user}\) has been %{CISCO ACTION:action} """ASA-7-710001, ASA-7-710002, ASA-7-710003, ASA-7-710005, ASA-7-710006""" CISCOFW710001_710002_710005_710006 %{WORD:protocol} (?:request|access) %{CISCO_ACTION:action} from %{IP:src_ip}/%{INT:src_port} to % {DATA:dst_interface}:%{IP:dst_ip}/%{INT:dst_port} """ASA-6-713172""" CISCOFW713172 Group = %{GREEDYDATA:group}, IP = %{IP:src_ip}, Automatic NAT Detection Status:\s+Remote end\s*%{DATA:is_remote_natted}\s*behind a NAT device\s+This\s+end\s*%{DATA:is_local_natted}\s*behind a NAT device """ASA-4-733100""" CISCOFW733100 \[\s*%{DATA:drop_type}\s*\] drop %{DATA:drop_rate_id} exceeded. Current burst rate is %{INT:drop_rate_current_burst} per second, max configured rate is %{INT:drop_rate_max_burst}; Current average rate is %{INT:drop_rate_current_avg} per second, max configured rate is %{I NT:drop_rate_max_avg}; Cumulative total count is %{INT:drop_total_count} """== End Cisco ASA ==""" """Shorewall firewall logs""" SHOREWALL (%{SYSLOGTIMESTAMP:timestamp}) (%{WORD:nf_host}) kernel:.*Shorewall:(%{WORD:nf_action1})?:(%{WORD:nf_action2})?.*IN=(% {USERNAME:nf_in_interface})?.*(OUT= *MAC=(%{COMMONMAC:nf_dst_mac}):(%{COMMONMAC:nf_src_mac})?|OUT=%{USERNAME:nf_out_interface}).*SRC=(% {IPV4:nf_src_ip}).*DST=(%{IPV4:nf_dst_ip}).*LEN=(%{WORD:nf_len}).?*TOS=(%{WORD:nf_tos}).?*PREC=(%{WORD:nf_prec}).?*TTL=(%{INT:nf_ttl}).?*IDE(%

Sample patterns for HAProxy

""" These patterns were tested w/ haproxy-1.4.15""" """ Documentation of the haproxy log formats can be found at the following links:""" """ http://code.google.com/p/haproxy-docs/wiki/HTTPLogFormat"" """ http://code.google.com/p/haproxy-docs/wiki/TCPLogFormat""" HAPROXYTIME (?!<[0-9])%{HOUR:haproxy_hour}:%{MINUTE:haproxy_minute}(?::%{SECOND:haproxy_second})(?![0-9]) HAPROXYDATE %{MONTHDAY:haproxy_monthday}/%{MONTH:haproxy_month}/%{YEAR:haproxy_year}:%{HAPROXYTIME:haproxy_time}.%{INT:haproxy_milliseconds} """ Override these default patterns to parse out what is captured in your haproxy.cfg""" HAPROXYCAPTUREDREQUESTHEADERS %{DATA:captured request headers} HAPROXYCAPTUREDRESPONSEHEADERS %{DATA:captured_response_headers} """ Example:""" """ These haproxy config lines will add data to the logs that are captured""" """ by the patterns below. Place them in your custom patterns directory to""" """ override the defaults.""" """ capture request header Host len 40""" """ capture request header X-Forwarded-For len 50""" """ capture request header Accept-Language len 50""" """ capture request header Referer len 200"" """ capture request header User-Agent len 200"" """ capture response header Content-Type len 30""" """ capture response header Content-Encoding len 10""" """ capture response header Cache-Control len 200" """ capture response header Last-Modified len 200""" """parse a haproxy 'httplog' line""" HAPROXYHTTPBASE %{IP:client_ip}:%{INT:client_port} \[%{HAPROXYDATE:accept_date}\] %{NOTSPACE:frontend_name} %{NOTSPACE:backend_name}/% {NOTSPACE:server_name) %{INT:time_request}/%{INT:time_queue}/%{INT:time_backend_connect}/%{INT:time_backend_response}/% {NOTSPACE:time duration} %{INT:http status code} %{NOTSPACE:bytes read} %{DATA:captured request cookie} %{DATA:captured response cookie} %{NOT SPACE:termination_state} %{INT:actconn}/%{INT:feconn}/%{INT:beconn}/%{INT:srvconn}/%{NOTSPACE:retries} %{INT:srv_queue}/%{INT:backend_queue} (\{%{HAPROXYCAPTUREDREQUESTHEADERS}\})?()?(\{%{HAPROXYCAPTUREDRESPONSEHEADERS}})?()?"(<BADREQ>|(%{WORD:http_verb})(% {URIPROTO:http_proto}://)?(?:%{USER:http_user}(?::[^@]*)?@)?(?:%{URIHOST:http_host})?(?:%{URIPATHPARAM:http_request})?(HTTP/% {NUMBER:http version})?))?" HAPROXYHTTP (?:%{SYSLOGTIMESTAMP:syslog_timestamp}|%{TIMESTAMP_IS08601:timestamp8601}) %{IPORHOST:syslog_server} %{SYSLOGPROG}: % {HAPROXYHTTPBASE} """parse a haproxy 'tcplog' line""" HAPROXYTCP (?:%{SYSLOGTIMESTAMP:syslog timestamp}|%{TIMESTAMP ISO8601:timestamp8601}) %{IPORHOST:syslog server} %{SYSLOGPROG}: % {IP:client_ip}:%{INT:client_port} \[%{HAPROXYDATE:accept_date}\] %{NOTSPACE:frontend_name} %{NOTSPACE:backend_name}/%{NOTSPACE:server_name} %{ INT:time_queue}/% [INT:time_backend_connect]/% [NOTSPACE:time_duration] % [NOTSPACE:bytes_read] % [NOTSPACE:termination_state} % [INT:actconn]/% [IN T:feconn}/%{INT:beconn}/%{INT:srvconn}/%{NOTSPACE:retries} %{INT:srv_queue}/%{INT:backend_queue}

Sample patterns for Java

JAVACLASS (?:[a-zA-Z\$_][a-zA-Z\$_0-9]*\.)*[a-zA-Z\$_][a-zA-Z\$_0-9]* """Space is an allowed character to match special cases like 'Native Method' or 'Unknown Source'""" JAVAFILE (?:[A-Za-z0-9_. -]+) """Allow special <init> method"" JAVAMETHOD (?:(<init>) |[a-zA-Z\$_][a-zA-Z\$_0-9]*) """Line number is optional in special cases 'Native method' or 'Unknown source'"" JAVASTACKTRACEPART %{SPACE}at %{JAVACLASS:class}\.%{JAVAMETHOD:method}\(%{JAVAFILE:file}(?::%{NUMBER:line})?\) """ Java Logs""" JAVATHREAD (?: [A-Z] {2}-Processor[\d]+) JAVACLASS (?: $[a-zA-Z0-9-]+\.)+[A-Za-z0-9\$]+$ JAVAFILE (?:[A-Za-z0-9 .-]+) JAVASTACKTRACEPART at %{JAVACLASS:class}\.%{WORD:method}\(%{JAVAFILE:file}:%{NUMBER:line}\) JAVALOGMESSAGE (.*) """ MMM dd, yyyy HH:mm:ss eg: Jan 9, 2014 7:13:13 AM""" CATALINA_DATESTAMP %{MONTH} %{MONTHDAY}, 20%{YEAR} %{HOUR}:?%{MINUTE}(?::?%{SECOND}) (?:AM|PM) """ yyyy-MM-dd HH:mm:ss,SSS ZZZ eg: 2014-01-09 17:32:25,527 -0800""" TOMCAT DATESTAMP 20% {YEAR} - % {MONTHNUM} - % {MONTHDAY} % {HOUR} : ?% {MINUTE} (?::?% {SECOND}) % {ISO8601 TIMEZONE} CATALINALOG %{CATALINA DATESTAMP:timestamp} %{JAVACLASS:class} %{JAVALOGMESSAGE:logmessage} """ 2014-01-09 20:03:28,269 -0800 | ERROR | com.example.service.ExampleService - something compeletely unexpected happened...""" TOMCATLOG %{TOMCAT_DATESTAMP:timestamp} \| %{LOGLEVEL:level} \| %{JAVACLASS:class} - %{JAVALOGMESSAGE:logmessage}

Sample patterns for Junos

"""JUNOS 11.4 RT FLOW patterns"""

RT_FLOW_EVENT (RT_FLOW_SESSION_CREATE|RT_FLOW_SESSION_CLOSE|RT_FLOW_SESSION_DENY)

RT_FLOW1 %{RT_FLOW_EVENT:event}: %{GREEDYDATA:close-reason}: %{IP:src-ip}/%{INT:src-port}->%{IP:dst-ip}/%{INT:dst-port} %{DATA:service} %{IP:n at-src-ip}/%{INT:nat-src-port}->%{IP:nat-dst-ip}/%{INT:nat-dst-port} %{DATA:src-nat-rule-name} %{DATA:dst-nat-rule-name} %{INT:protocol-id} %{ DATA:policy-name} %{DATA:from-zone} %{DATA:to-zone} %{INT:session-id} \d+\(%{DATA:sent}\) \d+\(%{DATA:received}\) %{INT:elapsed-time}.*

RT_FLOW2 %{RT_FLOW_EVENT:event}: session created %{IP:src-ip}/%{INT:src-port}->%{IP:dst-ip}/%{INT:dst-port} %{DATA:service} %{IP:nat-src-ip}/% {INT:nat-src-port}->%{IP:nat-dst-ip}/%{INT:nat-dst-port} %{DATA:src-nat-rule-name} %{DATA:dst-nat-rule-name} %{INT:protocol-id} %{DATA:policyname} %{DATA:from-zone} %{DATA:to-zone} %{INT:session-id} .*

RT_FLOW3 %{RT_FLOW_EVENT:event}: session denied %{IP:src-ip}/%{INT:src-port}->%{IP:dst-ip}/%{INT:dst-port} %{DATA:service} %{INT:protocol-id}\((\d\) %{DATA:policy-name} %{DATA:from-zone} %{DATA:to-zone} .*

Sample patterns for Linux Syslog

SYSLOG5424PRINTASCII [!-~]+

SYSLOGEASE2 (?:%{SYSLOGTIMESTAMP:timestamp}|%{TIMESTAMP_ISO8601:timestamp8601}) (?:%{SYSLOGFACILITY})?%{SYSLOGHOST:logsource}+(?: %{SYSLOGPROG}}:))

SYSLOGPAMSESSION %{SYSLOGBASE} (?=%{GREEDYDATA:message})%{WORD:pam_module}\(%{DATA:pam_caller}\): session %{WORD:pam_session_state} for user % {USERNAME:username}(?: by %{GREEDYDATA:pam_by})?

CRON_ACTION [A-Z]+

CRONLOG %{SYSLOGBASE} \(%{USER:user}\) %{CRON_ACTION:action} \(%{DATA:message}\)

SYSLOGLINE %{SYSLOGBASE2} %{GREEDYDATA:message}

"""IETF 5424 syslog(8) format (see http://www.rfc-editor.org/info/rfc5424)"""
SYSLOG5424PRI <%{NONNEGINT:syslog5424_pri}>
SYSLOG5424SD \{%{DATA}}\}
SYSLOG5424BASE %{SYSLOG5424PRI}%{NONNEGINT:syslog5424_ver} +{?:%{TIMESTAMP_ISO8601:syslog5424_ts}|-) +(?:%{HOSTNAME:syslog5424_host}|-) +(-|%{SYSLOG5424BASE %{SYSLOG5424PRINTASCII:syslog5424_most}|-) +(-|%{SYSLOG5424PRINTASCII:syslog5424_most}|-) +(?:%{HOSTNAME:syslog5424_most}|-) +(?:%{SYSLOG5424PRINTASCII:syslog5424_most}) +(?:%{SYSLOG5424PRINTASCII:syslog542

SYSLOG5424LINE %{SYSLOG5424BASE} +%{GREEDYDATA:syslog5424 msg}

Sample patterns for Mcollective

"""Remember, these can be multi-line events."""
MCOLLECTIVE ., \[%{TIMESTAMP_ISO8601:timestamp} #%{POSINT:pid}\]%{SPACE}%{LOGLEVEL:event_level}

MCOLLECTIVEAUDIT %{TIMESTAMP_ISO8601:timestamp}:

Sample patterns for MongoDB

MONGO_LOG %{SYSLOGTIMESTAMP:timestamp} \[%{WORD:component}]} %{GREEDYDATA:message} MONGO_QUERY \{ (?<= \).*(?= \)ntoreturn:) \} MONGO_SLOWQUERY %{WORD} %{MONGO_WORDDASH:database}\.%{MONGO_WORDDASH:collection} %{WORD}: %{MONGO_QUERY:query} %{WORD}:%{NONNEGINT:ntoreturn} %{WORD}:%{NONNEGINT:ntoskip} %{WORD}:%{NONNEGINT:nscanned}.*nreturned:%{NONNEGINT:nreturned}...+ (?<duration>[0-9]+)ms MONGO_WORDDASH \b[\w-]+\b MONGO3_SEVERITY \w MONGO3_COMPONENT %{WORD}!-MONGO3_COMPONENT %{WORD}!-MONGO3_LOG %{TIMESTAMP_ISO8601:timestamp} %{MONGO3_SEVERITY:severity} %{MONGO3_COMPONENT:component}%{SPACE}{?:\[%{DATA:context}\]}? % {GREEDYDATA:message}

Sample patterns for Nagios

NAGIOSTIME \[%{NUMBER:nagios_epoch}\] """nagios log types""" NAGIOS_TYPE_CURRENT_SERVICE_STATE CURRENT SERVICE STATE NAGIOS_TYPE_CURRENT_HOST_STATE CURRENT HOST STATE

NAGIOS_TYPE_SERVICE_NOTIFICATION SERVICE NOTIFICATION NAGIOS_TYPE_HOST_NOTIFICATION HOST NOTIFICATION

NAGIOS_TYPE_SERVICE_ALERT SERVICE ALERT NAGIOS TYPE HOST ALERT HOST ALERT

NAGIOS_TYPE_SERVICE_FLAPPING_ALERT SERVICE FLAPPING ALERT NAGIOS_TYPE_HOST_FLAPPING_ALERT HOST FLAPPING ALERT

NAGIOS_TYPE_SERVICE_DOWNTIME_ALERT SERVICE DOWNTIME ALERT NAGIOS_TYPE_HOST_DOWNTIME_ALERT HOST DOWNTIME ALERT

NAGIOS_TYPE_PASSIVE_SERVICE_CHECK PASSIVE SERVICE CHECK NAGIOS_TYPE_PASSIVE_HOST_CHECK PASSIVE HOST CHECK

NAGIOS_TYPE_SERVICE_EVENT_HANDLER SERVICE EVENT HANDLER NAGIOS TYPE HOST EVENT HANDLER HOST EVENT HANDLER

NAGIOS_TYPE_EXTERNAL_COMMAND EXTERNAL COMMAND NAGIOS_TYPE_TIMEPERIOD_TRANSITION TIMEPERIOD TRANSITION """external check types""" NAGIOS_EC_DISABLE_SVC_CHECK DISABLE_SVC_CHECK NAGIOS_EC_ENABLE_SVC_CHECK DISABLE_SVC_CHECK NAGIOS_EC_ENABLE_HOST_CHECK ENABLE_HOST_CHECK NAGIOS_EC_PROCESS_SERVICE_CHECK_RESULT PROCESS_SERVICE_CHECK_RESULT NAGIOS_EC_PROCESS_HOST_CHECK_RESULT PROCESS_HOST_CHECK_RESULT NAGIOS_EC_SCHEDULE_SERVICE_OWNTIME SCHEDULE_SERVICE_OWNTIME NAGIOS_EC_SCHEDULE_HOST_DVMITIME SCHEDULE_HOST_DOWNTIME NAGIOS_EC_DISABLE_HOST_SVC_NOTIFICATIONS DISABLE_HOST_SVC_NOTIFICATIONS NAGIOS_EC_DISABLE_HOST_NOTIFICATIONS DISABLE_HOST_SVC_NOTIFICATIONS NAGIOS_EC_DISABLE_HOST_NOTIFICATIONS DISABLE_HOST_NOTIFICATIONS NAGIOS_EC_ENABLE_HOST_NOTIFICATIONS DISABLE_HOST_NOTIFICATIONS NAGIOS_EC_ENABLE_HOST_NOTIFICATIONS DISABLE_HOST_NOTIFICATIONS NAGIOS_EC_ENABLE_SVC_NOTIFICATIONS DISABLE_SVC_NOTIFICATIONS NAGIOS_EC_ENABLE_SVC_NOTIFICATIONS DISABLE_SVC_NOTIFICATIONS NAGIOS_EC_ENABLE_SVC_NOTIFICATIONS DISABLE_SVC_NOTIFICATIONS NAGIOS_EC_ENABLE_SVC_NOTIFICATIONS DISABLE_SVC_NOTIFICATIONS NAGIOS_EC_ENABLE_SVC_NOTIFICATIONS DISABLE_SVC_NOTIFICATIONS

NAGIOS WARNING Warning:%{SPACE}%{GREEDYDATA:nagios message}

NAGIOS_CURRENT_SERVICE_STATE %{NAGIOS_TYPE_CURRENT_SERVICE_STATE:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_service};% {DATA:nagios_state};%{DATA:nagios_statetype};%{DATA:nagios_statecode};%{GREEDYDATA:nagios_message} NAGIOS_CURRENT_HOST_STATE %{NAGIOS_TYPE_CURRENT_HOST_STATE:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_state};% {DATA:nagios_statetype};%{DATA:nagios_statecode};%{GREEDYDATA:nagios_message}

NAGIOS_SERVICE_NOTIFICATION %{NAGIOS_TYPE_SERVICE_NOTIFICATION:nagios_type}: %{DATA:nagios_notifyname};%{DATA:nagios_hostname};% {DATA:nagios_service};%{DATA:nagios_state};%{DATA:nagios_contact};%{GREEDYDATA:nagios_message} NAGIOS_HOST_NOTIFICATION %{NAGIOS_TYPE_HOST_NOTIFICATION:nagios_type}: %{DATA:nagios_notifyname};%{DATA:nagios_hostname};% {DATA:nagios_state};%{DATA:nagios_contact};%{GREEDYDATA:nagios_message}

NAGIOS_SERVICE_ALERT %{NAGIOS_TYPE_SERVICE_ALERT:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_service};%{DATA:nagios_state};% {DATA:nagios_statelevel};%{NUMBER:nagios_attempt};%{GREEDYDATA:nagios_message} NAGIOS_HOST_ALERT %{NAGIOS_TYPE_HOST_ALERT:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_state};%{DATA:nagios_statelevel};% {NUMBER:nagios_attempt};%{GREEDYDATA:nagios_message}

NAGIOS_SERVICE_FLAPPING_ALERT %{NAGIOS_TYPE_SERVICE_FLAPPING_ALERT:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_service};% {DATA:nagios_state};%{GREEDYDATA:nagios_message} NAGIOS_HOST_FLAPPING_ALERT %{NAGIOS_TYPE_HOST_FLAPPING_ALERT:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_state};% {GREEDYDATA:nagios_message}

NAGIOS_SERVICE_DOWNTIME_ALERT %{NAGIOS_TYPE_SERVICE_DOWNTIME_ALERT:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_service};% {DATA:nagios_state};%{(REEDYDATA:nagios_comment) NAGIOS_HOST_DOWNTIME_ALERT %{NAGIOS_TYPE_HOST_DOWNTIME_ALERT:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_state};% {REEDYDATA:nagios_comment}

NAGIOS_PASSIVE_SERVICE_CHECK %{NAGIOS_TYPE_PASSIVE_SERVICE_CHECK:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_service};% {DATA:nagios_state};%{GREEDYDATA:nagios_comment} NAGIOS_PASSIVE_HOST_CHECK %{NAGIOS_TYPE_PASSIVE_HOST_CHECK:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_state};% {GREEDYDATA:nagios_comment}

NAGIOS_SERVICE_EVENT_HANDLER %{NAGIOS_TYPE_SERVICE_EVENT_HANDLER:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_service};% {DATA:nagios_state};%{DATA:nagios_statelevel};%{DATA:nagios_event_handler_name} NAGIOS_HOST_EVENT_HANDLER %{NAGIOS_TYPE_HOST_EVENT_HANDLER:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_state};% {DATA:nagios_statelevel};%{DATA:nagios_event_handler_name}

NAGIOS_TIMEPERIOD_TRANSITION %{NAGIOS_TYPE_TIMEPERIOD_TRANSITION:nagios_type}: %{DATA:nagios_service};%{DATA:nagios_unknown1};% {DATA:nagios_unknown2}

"""Disable host & service check"""

NAGIOS_EC_LINE_DISABLE_SVC_CHECK %{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_DISABLE_SVC_CHECK:nagios_command};% {DATA:nagios_hostname};%{DATA:nagios_service} NAGIOS_EC_LINE_DISABLE_HOST_CHECK %{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_DISABLE_HOST_CHECK:nagios_command};% {DATA:nagios_hostname}

"""Enable host & service check"""

{DATA:nagios hostname}

NAGIOS_EC_LINE_ENABLE_SVC_CHECK %{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_ENABLE_SVC_CHECK:nagios_command};% {DATA:nagios_hostname};%{DATA:nagios_service} NAGIOS_EC_LINE_ENABLE_HOST_CHECK %{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_ENABLE_HOST_CHECK:nagios_command};%

"""Process host & service check"""

NAGIOS EC LINE PROCESS SERVICE CHECK RESULT %{NAGIOS TYPE EXTERNAL COMMAND:nagios type}: %

{NAGIOS_EC_PROCESS_SERVICE_CHECK_RESULT:nagios_command};%{DATA:nagios_hostname};%{DATA:nagios_service};%{DATA:nagios_state};% {GREEDYDATA:nagios_check_result}

NAGIOS_EC_LINE_PROCESS_HOST_CHECK_RESULT %{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_PROCESS_HOST_CHECK_RESULT:nagios_command};%{DATA:nagios hostname};%{DATA:nagios state};%{GREEDYDATA:nagios check result}

"""Disable host & service notifications"""

NAGIOS_EC_LINE_DISABLE_HOST_SVC_NOTIFICATIONS %{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %

{NAGIOS_EC_DISABLE_HOST_SVC_NOTIFICATIONS:nagios_command};%{GREEDYDATA:nagios_hostname}

NAGIOS_EC_LINE_DISABLE_HOST_NOTIFICATIONS %{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %

{NAGIOS_EC_DISABLE_HOST_NOTIFICATIONS:nagios_command};%{GREEDYDATA:nagios_hostname}

NAGIOS_EC_LINE_DISABLE_SVC_NOTIFICATIONS %{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_DISABLE_SVC_NOTIFICATIONS:nagios_command};%{ DATA:nagios_hostname};%{GREEDYDATA:nagios_service}

"""Enable host & service notifications"""

NAGIOS EC LINE ENABLE HOST SVC NOTIFICATIONS %{NAGIOS TYPE EXTERNAL COMMAND:nagios type}: %

{NAGIOS_EC_ENABLE_HOST_SVC_NOTIFICATIONS:nagios_command};%{GREEDYDATA:nagios_hostname}

NAGIOS_EC_LINE_ENABLE_HOST_NOTIFICATIONS %{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_ENABLE_HOST_NOTIFICATIONS:nagios_command};%{ GREEDYDATA:nagios_hostname}

NAGIOS_EC_LINE_ENABLE_SVC_NOTIFICATIONS %{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_ENABLE_SVC_NOTIFICATIONS:nagios_command};%{DA TA:nagios_hostname};%{GREEDYDATA:nagios_service}

"""Schedule host & service downtime"""

NAGIOS_EC_LINE_SCHEDULE_HOST_DOWNTIME %{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_SCHEDULE_HOST_DOWNTIME:nagios_command};% {DATA:nagios_hostname};%{NUMBER:nagios_start_time};%{NUMBER:nagios_end_time};%{NUMBER:nagios_fixed};%{NUMBER:nagios_trigger_id};% {NUMBER:nagios_duration};%{DATA:author};%{DATA:comment}

"""End matching line"""

NAGIOSLOGLINE %{NAGIOSTIME} (?:%{NAGIOS_NARNING}|%{NAGIOS_CURRENT_SERVICE_STATE}|%{NAGIOS_CURRENT_HOST_STATE}}% {NAGIOS_SERVICE_NOTIFICATION}}%{NAGIOS_HOST_NOTIFICATION}}%{NAGIOS_CURRENT_SERVICE_ALERT}}%{NAGIOS_CURRENT_HOST_SERVICE_FLAPPING_ALERT}}% {NAGIOS_FILAPPING_ALERT}}%{NAGIOS_SERVICE_DOWNTIME_ALERT}}%{NAGIOS_HOST_DOWNTIME_ALERT}}%{NAGIOS_PASSIVE_SERVICE_CHECK}}% {NAGIOS_PASSIVE_HOST_CHECK}%{NAGIOS_SERVICE_DOWNTIME_ALERT}}%{NAGIOS_HOST_DOWNTIME_ALERT}}%{NAGIOS_PASSIVE_SERVICE_CHECK}}% {NAGIOS_CC_LINE_DISABLE_SVC_CHECK}%{NAGIOS_EC_LINE_ENABLE_SVC_CHECK}%{NAGIOS_EC_LINE_DISABLE_HOST_CHECK}}% {NAGIOS_CC_LINE_SCHEDULE_HOST_CHECK}%{NAGIOS_EC_LINE_PROCESS_HOST_CHECK_RESULT}%{NAGIOS_EC_LINE_PROCESS_SERVICE_CHECK_RESULT}}% {NAGIOS_EC_LINE_SCHEDULE_HOST_DOWNTIME}}%{NAGIOS_EC_LINE_PROCESS_HOST_CHECK_RESULT}%{NAGIOS_EC_LINE_PROCESS_SERVICE_CHECK_RESULT}% {NAGIOS_EC_LINE_SCHEDULE_HOST_DOWNTIME}}%{NAGIOS_EC_LINE_PROCESS_NOT_FICATIONS}% {NAGIOS_EC_LINE_SCHEDULE_HOST_DOWNTIME}}%{NAGIOS_EC_LINE_PROCESS_NOTIFICATIONS}% {NAGIOS_EC_LINE_SCHEDULE_HOST_NOTIFICATIONS}% {NAGIOS_EC_LINE_ENABLE_HOST_NOTIFICATIONS}% {NAGIOS_EC_LINE_ENABLE_SVC_NOTIFICATIONS}% {NAGIOS_EC_LINE_ENABLE_SVC_NOTIFICATIONS}%

Sample patterns for PostgreSQL

"""Default postgresql pg_log format pattern""" POSTGRESQL %{DATESTAMP:timestamp} %{TZ} %{DATA:user_id} %{GREEDYDATA:connection_id} %{POSINT:pid}

Sample patterns for Rails

RUUID \h{32}
"""rails controller with action"""
RCONTROLLER (?<controller>[^#]+)#(?<action>\w+)
"""this will often be the only line:"""
RAILS3HEAD (?m)Started %(WORD:verb) "%(URIPATHPARAM:request)" for %(IPORHOST:clientip) at (?<timestamp>%(YEAR)-%(MONTHNUM)-%(MONTHDAY) %
{HOUR):%(MINUTE):%(SECOND) %(ISO8601_TIMEZONE))
"""for some a strange reason, params are stripped of (} - not sure that's a good idea."""
RPROCESSING \W*Processing by %(RCONTROLLER) as (?<format>\S+) (?:\W*Parameters: (%(DATA:params})\W*)?
RAILS3FROOT Completed %(NUMBER:response)%(DATA) in %(NUMBER:totalms)ms %(RAILS3PROFILE)%(GREEDYDATA)
RAILS3FROFILE (?:\Views: %(NUMBER:viewms)ms \| ActiveRecord: %(NUMBER:activerecordms)ms)\(ActiveRecord: %(NUMBER:activerecordms)ms)?
"""putting it all together"""
RAILS3 %(RAILS3HEAD) (?:%(REPROCESSING))?(?<context>(?:%(DATA)\n)*)(?:%(RAILS3FOOT))?

Sample patterns for Redis

REDISTIMESTAMP %{MONTHDAY} %{MONTH} %{TIME} REDISLOG \[%{POSINT:pid}\] %{REDISTIMESTAMP:timestamp} *

Sample patterns for Ruby

RUBY_LOGLEVEL (?:DEBUG|FATAL|ERROR|WARN|INFO) RUBY_LOGGER [DFEWI], \[%{TIMESTAMP_ISO8601:timestamp} #%{POSINT:pid}\] *%{RUBY_LOGLEVEL:loglevel} -- +%{DATA:progname}: %{GREEDYDATA:message}

4.5.8.8.6. JMESPath syntax

This topic describes the common JMESPath syntax and provides examples on how to use the syntax.

JMESPath is an enhanced query and computing language for JSON. You can use JMESPath to extract, compute, and convert JSON data. For more information, see JMESPath Tutorial.

The data transformation feature supports the json_select , e_json , and e_split functions. You can use JMESPath in the functions to extract the values of fields or JSON expressions or compute specific values. Examples:

json_select(Value, "JMESPath expression", ...)

e_json(Field name, jmes="JMESPath expression", ...)
e_split(Field name, ... jmes="JMESPath expression", ...)

For more information about how to use the functions, see json_select, e_json, and e_split.

Obtain a value by using the key of a field

Raw log

json data: {"a":"foo","b":"bar","c":"baz"}

Transformation rule

 $\ensuremath{\#}$ Obtain the value of a from the JSON expression.

e_set("a1", json_select(v("json_data"), "a"))
Obtain the value of b from the JSON expression.

e_set("b1", json_select(v("json_data"), "b"))

Obtain the value of c from the JSON expression.

e_set("c1", json_select(v("json_data"), "c"))

• Result

```
al:foo
bl:bar
cl:baz
json_data:{"a":"foo","b":"bar","c":"baz"}
```

Obtain a nested value

Raw log

json_data:{"a": {"b":{"c":{"d":"value"}}}

• Transformation rule

Obtain the value of d from the JSON expression.
e_set("e", json_select(v("json_data"), "a.b.c.d"))

• Result

e:value
json_data:{"a": {"b":{"c":{"d":"value"}}}

Obtain a value by slicing data

Raw log

json_data:{"a": ["b", "c", "d", "e", "f"]}

• Transformation rule

Obtain the value of the a field starting from slice 2. The value at slice 0 is b, and the value at slice 1 is c.
e_set("key", json_select(v("json_data"), "a[2:]"))

• Result

json_data:{"a": ["b", "c", "d", "e", "f"]}
key:["d", "e", "f"]

Obtain a value by combining the preceding methods

Raw log

json_data:{"a": {"b": {"c": [{"d": [0, [1, 2]]}, {"d": [3, 4]}]}}

• Transformation rule

c[0] specifies {"d": [0, [1, 2]]}. d[1] specifies [1, 2]]. The return value is 1. e_set("key", json_select(v("json_data"), "a.b.c[0].d[1][0]"))

Result

json_data:{"a": {"b": {"c": [{"d": [0, [1, 2]]}, {"d": [3, 4]}]}}
key:1

Obtain a value by using projection

• Example 1

◦ Raw log

json_data:{"people": [{"first": "James", "last": "d"},{"first": "Jacob", "last": "e"},{"first": "Jayden", "last": "f"},{"missing": "different"}],"foo": {"bar": "baz"}}

• Transformation rule

Obtain the values of the first fields from the people list. e_set("key", json_select(v("json_data"), "people[*].first"))

Result

json_data:{"people": [{"first": "James", "last": "d"},{"first": "Jacob", "last": "e"},{"first": "Jayden", "last": "f"},{"missing": "different"}],"foo": {"bar": "baz"}} key:["James", "Jacob", "Jayden"]

• Example 2

Raw log

json_data:{"ops": {"functionA": {"numArgs": 2},"functionB": {"numArgs": 3},"functionC": {"variadic": true}}}

• Transformation rule

Obtain the values of the numArgs fields from the ops list. e_set("key", json_select(v("json_data"), "ops.*.numArgs"))

Result

json_data:{"ops": {"functionA": {"numArgs": 2},"functionB": {"numArgs": 3},"functionC": {"variadic": true}}}
key:[2, 3]

• Example 3

Raw log

json_data:{"machines": [{"name": "a", "state": "running"},{"name": "b", "state": "stopped"},{"name": "c", "state": "running"}]}

Transformation rule

Obtain the values of the name fields for which the state field is running from the machines list. e_set("key", json_select(v("json_data"), "machines[?state=='running'].name"))

Result

json_data:{"machines": [{"name": "a", "state": "running"},{"name": "b", "state": "stopped"},{"name": "c", "state": "running"}]}
key:["a", "c"]

Extract multiple values

Raw log

json_data:{"people": [{"name": "a","state": {"name": "up"}}, {"name": "b","state": {"name": "down"}}]}

Transformation rule

Obtain the values of the name and state fields from the people list. e_set("key", json_select(v("json_data"), "people[].[name, state.name]"))

Result

json_data:{"people": [{"name": "a","state": {"name": "up"}},{"name": "b","state": {"name": "down"}}]}
key:[["a", "up"], ["b", "down"]]

Calculate the number of elements in an array

Raw log

json_data:{"a": ["b", "c", "d", "e", "f"]}

Transformation rule

Obtain the number of elements in the a array. e_set("key", json_select(v("json_data"), "length(a)"))

Obtain the number of elements in the a array. If the result of length(a) is greater than 0, set the no-empty parameter to true. e_if(json_select(v("json_data"), "length(a)", default=0), e_set("no-empty", true))

• Result

json_data:{"a": ["b", "c", "d", "e", "f"]}
key:5
json_data:{"a": ["b", "c", "d", "e", "f"]}

no-empty:true

4.5.8.8.7. Date and time formatting directives

The ANSI C standard defines a set of directives used to parse and format date and time strings.

Log Service supports all the directives defined in the ANSI C (1989 version) standard. The following table describes these directives and provides specific examples and notes.

Directive	Description	Example	Note
%a	The abbreviation of the weekday.	Sun,, Mon	Date and time strings are displayed in the en-US locale. Other locales are not supported.
%A	The full name of the weekday.	Sunday,, Monday	Date and time strings are displayed in the en-US locale. Other locales are not supported.
%w	The weekday represented as a decimal number. The value 0 indicates Sunday. The value 6 indicates Saturday.	0, 1, 2, 3, 4, 5, 6	None.
%d	The day of the month represented as a zero- padded decimal number.	01, 02,, 31	The leading zero is optional for formatting directives %d, %m, %H, %I, %M, %S, %J, %U, %W, %V, %y when they are used to parse date and time strings.
%b	The abbreviation of the month.	Jan, Feb,, Dec	Date and time strings are displayed in the en-US locale. Other locales are not supported.
%B	The full name of the month.	January, February,, December	The leading zero is optional for formatting directives %d, %m, %H, %I, %M, %S, %J, %U, %W, %V, %y when they are used to parse date and time strings.
%m	The month represented as a zero-padded decimal number.	01, 02,, 12	The leading zero is optional for formatting directives %d, %m, %H, %I, %M, %S, %J, %U, %W, %V, %y when they are used to parse date and time strings.
%у	The year without the century part, represented as a zero-padded decimal number.	00, 01,, 99	The leading zero is optional for formatting directives %d, %m, %H, %I, %M, %S, %J, %U, %W, %V, %y when they are used to parse date and time strings.

%Y	The year with the century part, represented as a zero-padded decimal number.	0001, 0002,, 2013, 2014,, 9998, 9999	A year can be parsed from a number that ranges from 1 to 9999. If the year is earlier than 1000, it must be zero-filled to 4-digit width. For example, 0180 indicates the year of 180 AD.
%Н	The hour in the 24-hour format, represented as a zero-padded decimal number.	00, 01,, 23	The leading zero is optional for formatting directives %d, %m, %H, %I, %M, %S, %J, %U, %W, %V, %y when they are used to parse date and time strings.
%I	The hour in the 12-hour format, represented as a zero-padded decimal number.	01, 02,, 12	The leading zero is optional for formatting directives %d, %m, %H, %I, %M, %S, %J, %U, %W, %V, %y when they are used to parse date and time strings.
%р	The period in the 12-hour format.	AM, PM	 Date and time strings are displayed in the en-US locale. Other locales are not supported. The sp directive affects the hour field in the parsing result when the si directive is used to parse the hour.
%M	The minute represented as a zero-padded decimal number.	00 ,01,, 59	The leading zero is optional for formatting directives %d, %m, %H, %I, %M, %S, %J, %U, %W, %V, %y when they are used to parse date and time strings.
%S	The second represented as a zero-padded decimal number.	00 ,01,, 59	 Leap seconds are not supported. The leading zero is optional for formatting directives %d, %m, %H, %I, %M, %S, %J, %U, %W, %V, %y when they are used to parse date and time strings.
%f	The microsecond represented as a zero- padded decimal number.	000000, 000001,, 999999	The %f directive can be used to parse the microsecond from a numeric string consisting of 0 to 6 characters.
%z	The UTC offset in the \pm HHMM[SS[.ffffff]] format. An empty string is generated for this directive during parsing if the date and time string does not contain the time zone information.	(empty), +0000, -0400, +1030, +063415, -030712.345216	The %z and %Z directives are replaced by empty strings during parsing if the date and time string does not contain the time zone information. The minute information is optional in the string to be parsed by the %z directive to the <code>HHHM[SS[.ffffff]]</code> format. Strings separated by colons (:) are supported during parsing. For example, <code>+01:00:00</code> is parsed as an offset of one hour. In addition, <code>Z</code> is equivalent to <code>+00:00</code> .
%Z	The name of the time zone. An empty string is generated for this directive during parsing if the date and time string does not contain the time zone information.	(empty), UTC, EST, CST	None.
%j	The day of the year.	001, 002,, 366	The leading zero is optional for formatting directives %d, %m, %H, %I, %M, %S, %J, %U, %W, %V, %y when they are used to parse date and time strings.
%U	The week number of the year. Sunday is the first day of each week. A day before the first Sunday of the year is regarded as a day in week 0.	00, 01,, 53	 When this directive is used to parse data and time strings, the values obtained by using the %U and %W directives can only be used in calculation. The leading zero is optional for formatting directives %d, %m, %H, %I, %M, %S, %J, %U, %W, %Y, %y when they are used to parse date and time strings.
%W	The week number of the year. Monday is the first day of each week. A day before the first Monday of the year is regarded as a day in week 0.	00, 01,, 53	 When this directive is used to parse data and time strings, the values obtained by using the %U and %W directives can only be used in calculation. The leading zero is optional for formatting directives %d, %m, %H, %I, %M, %S, %J, %U, %W, %Y, %y when they are used to parse date and time strings.
%с	The date and time representation in the current locale.	Tue Aug 16 21:30:00 1988	Date and time strings are displayed in the en-US locale. Other locales are not supported.
%x	The date representation in the current locale.	08/16/88	Date and time strings are displayed in the en-US locale. Other locales are not supported.
%X	The time representation in the current locale.	21:30:00	Date and time strings are displayed in the en-US locale. Other locales are not supported.
%%	The literal % character.	8	None.

Several additional directives not defined in the C (1989 version) standard are included for convenience. The following table describes the directives.

Directive	Description	Example	Note
%G	The ISO 8601 week-based year.	0001, 0002,, 2013, 2014,, 9998, 9999	When this directive is used to parse data and time strings, the values obtained by using the sv directive can only be used in calculation.
%u	The day of the week in the ISO 8601 format. Monday is the first day of each week.	1, 2,, 7	None.
%V	The week number of the year in the ISO 8601 format. Monday is the first day of each week.	01, 02,, 53	 When this directive is used to parse data and time strings, the values obtained by using the %v directive can only be used in calculation. The leading zero is optional for formatting directives %d, %m, %H, %I, %M, %S, %J, %U, %W, %V, %y when they are used to parse date and time strings.

4.5.8.8.8. Time zones

This topic describes all valid values of the tz parameter that is used in the date and time functions.

The domain specific language (DSL) syntax supports multiple date and time functions. Each function uses tz=Time zone string to provide time zone information. Examples:

dt_parse(v("__time__"), tz="Asia/Shanghai") # Parse the __time__ field to the time in the time zone of Shanghai. dt_now(tz="Asia/Tokyo") # Obtain the current time in the time zone of Tokyo.

The following code contains all valid values of the tz parameter. Each line displays four time zone strings. The strings are separated by commas (,).

Africa/Abidjan, Africa/Accra, Africa/Addis Ababa, Africa/Algiers Africa/Asmara, Africa/Asmera, Africa/Bamako, Africa/Bangui Africa/Banjul, Africa/Bissau, Africa/Blantyre, Africa/Brazzaville Africa/Bujumbura, Africa/Cairo, Africa/Casablanca, Africa/Ceuta Africa/Conakry, Africa/Dakar, Africa/Dar es Salaam, Africa/Djibouti Africa/Douala, Africa/El_Aaiun, Africa/Freetown, Africa/Gaborone Africa/Harare, Africa/Johannesburg, Africa/Juba, Africa/Kampala Africa/Khartoum, Africa/Kigali, Africa/Kinshasa, Africa/Lagos Africa/Libreville, Africa/Lome, Africa/Luanda, Africa/Lubumbashi Africa/Lusaka, Africa/Malabo, Africa/Maputo, Africa/Maseru Africa/Mbabane, Africa/Mogadishu, Africa/Monrovia, Africa/Nairobi Africa/Ndjamena, Africa/Niamey, Africa/Nouakchott, Africa/Ouagadougou Africa/Porto-Novo, Africa/Sao_Tome, Africa/Timbuktu, Africa/Tripoli Africa/Tunis, Africa/Windhoek, America/Adak, America/Anchorage America/Anguilla, America/Antigua, America/Araguaina, America/Argentina/Buenos Aires America/Argentina/Catamarca, America/Argentina/ComodRivadavia, America/Argentina/Cordoba, America/Argentina/Jujuy America/Argentina/La_Rioja, America/Argentina/Mendoza, America/Argentina/Rio_Gallegos, America/Argentina/Salt America/Argentina/San_Juan, America/Argentina/San_Luis, America/Argentina/Tucuman, America/Argentina/Ushuaia America/Aruba, America/Asuncion, America/Atikokan, America/Atka America/Bahia, America/Bahia Banderas, America/Barbados, America/Belem America/Belize, America/Blanc-Sablon, America/Boa_Vista, America/Bogota America/Boise, America/Buenos_Aires, America/Cambridge_Bay, America/Campo_Grande America/Cancun, America/Caracas, America/Catamarca, America/Caye America/Cayman, America/Chicago, America/Chihuahua, America/Coral Harbour America/Cordoba, America/Costa Rica, America/Creston, America/Cuiaba America/Curacao, America/Danmarkshavn, America/Dawson, America/Dawson_Creek America/Denver, America/Detroit, America/Dominica, America/Edmonton America/Eirunepe, America/El_Salvador, America/Ensenada, America/Fort_Nelson America/Fort_Wayne, America/Fortaleza, America/Glace_Bay, America/Godthab America/Goose Bay, America/Grand Turk, America/Grenada, America/Guadeloupe America/Guatemala, America/Guayaquil, America/Guyana, America/Halifax America/Havana, America/Hermosillo, America/Indiana/Indianapolis, America/Indiana/Knox America/Indiana/Marengo, America/Indiana/Petersburg, America/Indiana/Tell_City, America/Indiana/Vevay America/Indiana/Vincennes, America/Indiana/Winamac, America/Indianapolis, America/Inuvik America/Iqaluit, America/Jamaica, America/Jujuy, America/Juneau America/Kentucky/Louisville, America/Kentucky/Monticello, America/Knox IN, America/Kralendijk America/La_Paz, America/Lima, America/Los_Angeles, America/Louisville America/Lower_Princes, America/Maceio, America/Managua, America/Manaus America/Marigot, America/Martinique, America/Matamoros, America/Mazatlan America/Mendoza, America/Menominee, America/Merida, America/Metlakatla America/Mexico City, America/Miguelon, America/Moncton, America/Monterrey America/Montevideo, America/Montreal, America/Montserrat, America/Nassa America/New_York, America/Nipigon, America/Nome, America/Noronha America/North_Dakota/Beulah, America/North_Dakota/Center, America/North_Dakota/New_Salem, America/Ojinaga America/Panama, America/Pangnirtung, America/Paramaribo, America/Phoenix America/Port-au-Prince, America/Port of Spain, America/Porto Acre, America/Porto Velho America/Puerto Rico, America/Punta Arenas, America/Rainy River, America/Rankin Inlet America/Recife, America/Regina, America/Resolute, America/Rio Bran America/Rosario, America/Santa_Isabel, America/Santarem, America/Santiago America/Santo_Domingo, America/Sao_Paulo, America/Scoresbysund, America/Shiprock America/Sitka, America/St_Barthelemy, America/St_Johns, America/St_Kitts America/St Lucia, America/St Thomas, America/St Vincent, America/Swift Current America/Tegucigalpa, America/Thule, America/Thunder_Bay, America/Tijuana America/Toronto, America/Tortola, America/Vancouver, America/Virgir America/Whitehorse, America/Winnipeg, America/Yakutat, America/Yellowknife

Antarctica/Casey, Antarctica/Davis, Antarctica/DumontDUrville, Antarctica/Macquarie Antarctica/Mawson, Antarctica/McMurdo, Antarctica/Palmer, Antarctica/Rother Antarctica/South Pole, Antarctica/Syowa, Antarctica/Troll, Antarctica/Vostok Arctic/Longyearbyen, Asia/Aden, Asia/Almaty, Asia/Amman Asia/Anadyr, Asia/Aqtau, Asia/Aqtobe, Asia/Ashgabat Asia/Ashkhabad, Asia/Atyrau, Asia/Baghdad, Asia/Bahrain Asia/Baku, Asia/Bangkok, Asia/Barnaul, Asia/Beirut Asia/Bishkek, Asia/Brunei, Asia/Calcutta, Asia/Chita Asia/Choibalsan, Asia/Chongging, Asia/Chungking, Asia/Colombo Asia/Dacca, Asia/Damascus, Asia/Dhaka, Asia/Dili Asia/Dubai, Asia/Dushanbe, Asia/Famagusta, Asia/Gaza Asia/Harbin, Asia/Hebron, Asia/Ho_Chi_Minh, Asia/Hong_Ko Asia/Hovd, Asia/Irkutsk, Asia/Istanbul, Asia/Jakarta Asia/Javapura, Asia/Jerusalem, Asia/Kabul, Asia/Kamchatka Asia/Karachi, Asia/Kashgar, Asia/Kathmandu, Asia/Katmandu Asia/Khandyga, Asia/Kolkata, Asia/Krasnoyarsk, Asia/Kuala_Lumpur Asia/Kuching, Asia/Kuwait, Asia/Macao, Asia/Macau Asia/Magadan, Asia/Makassar, Asia/Manila, Asia/Muscat Asia/Nicosia, Asia/Novokuznetsk, Asia/Novosibirsk, Asia/Omsk Asia/Oral, Asia/Phnom Penh, Asia/Pontianak, Asia/Pyongyang Asia/Qatar, Asia/Qostanay, Asia/Qyzylorda, Asia/Rangoon Asia/Riyadh, Asia/Saigon, Asia/Sakhalin, Asia/Samarkand Asia/Seoul, Asia/Shanghai, Asia/Singapore, Asia/Srednekolymsk Asia/Taipei, Asia/Tashkent, Asia/Tbilisi, Asia/Tehran Asia/Tel Aviv, Asia/Thimbu, Asia/Thimphu, Asia/Tokyo Asia/Tomsk, Asia/Ujung_Pandang, Asia/Ulaanbaatar, Asia/Ulan_Bator Asia/Urumqi, Asia/Ust-Nera, Asia/Vientiane, Asia/Vladivostok Asia/Yakutsk, Asia/Yangon, Asia/Yekaterinburg, Asia/Yerevan Atlantic/Azores, Atlantic/Bermuda, Atlantic/Canary, Atlantic/Cape Verde Atlantic/Faeroe, Atlantic/Faroe, Atlantic/Jan Mayen, Atlantic/Madeira Atlantic/Reykjavik, Atlantic/South_Georgia, Atlantic/St_Helena, Atlantic/Stanley Australia/ACT, Australia/Adelaide, Australia/Brisbane, Australia/Broken_Hill Australia/Canberra, Australia/Currie, Australia/Darwin, Australia/Eucla Australia/Hobart, Australia/LHI, Australia/Lindeman, Australia/Lord How Australia/Melbourne, Australia/NSW, Australia/North, Australia/Perth Australia/Queensland, Australia/South, Australia/Sydney, Australia/Tasmania Australia/Victoria, Australia/West, Australia/Yancowinna, Brazil/Acre Brazil/DeNoronha, Brazil/East, Brazil/West, CET CST6CDT, Canada/Atlantic, Canada/Central, Canada/Eastern Canada/Mountain, Canada/Newfoundland, Canada/Pacific, Canada/Saskatchewan Canada/Yukon, Chile/Continental, Chile/EasterIsland, Cuba EET, EST, EST5EDT, Egypt Eire, Etc/GMT, Etc/GMT+0, Etc/GMT+1 Etc/GMT+10, Etc/GMT+11, Etc/GMT+12, Etc/GMT+2 Etc/GMT+3, Etc/GMT+4, Etc/GMT+5, Etc/GMT+6 Etc/GMT+7, Etc/GMT+8, Etc/GMT+9, Etc/GMT-0 Etc/GMT-1, Etc/GMT-10, Etc/GMT-11, Etc/GMT-12 Etc/GMT-13, Etc/GMT-14, Etc/GMT-2, Etc/GMT-3 Etc/GMT-4, Etc/GMT-5, Etc/GMT-6, Etc/GMT-7 Etc/GMT-8, Etc/GMT-9, Etc/GMT0, Etc/Greenwich Etc/UCT, Etc/UTC, Etc/Universal, Etc/Zulu Europe/Amsterdam, Europe/Andorra, Europe/Astrakhan, Europe/Athens Europe/Belfast, Europe/Belgrade, Europe/Berlin, Europe/Bratislava Europe/Brussels, Europe/Bucharest, Europe/Budapest, Europe/Businge Europe/Chisinau, Europe/Copenhagen, Europe/Dublin, Europe/Gibraltar Europe/Guernsey, Europe/Helsinki, Europe/Isle of Man, Europe/Istanbul Europe/Jersey, Europe/Kaliningrad, Europe/Kiev, Europe/Kirov Europe/Lisbon, Europe/Ljubljana, Europe/London, Europe/Luxem Europe/Madrid, Europe/Malta, Europe/Mariehamn, Europe/Minsk Europe/Monaco, Europe/Moscow, Europe/Nicosia, Europe/Oslo Europe/Paris, Europe/Podgorica, Europe/Prague, Europe/Riga Europe/Rome, Europe/Samara, Europe/San_Marino, Europe/Sarajevo Europe/Saratov, Europe/Simferopol, Europe/Skopje, Europe/Sofia Europe/Stockholm, Europe/Tallinn, Europe/Tirane, Europe/Tiraspol Europe/Ulyanovsk, Europe/Uzhgorod, Europe/Vaduz, Europe/Vatican Europe/Vienna, Europe/Vilnius, Europe/Volgograd, Europe/Warsaw Europe/Zagreb, Europe/Zaporozhye, Europe/Zurich, GB GB-Eire, GMT, GMT+0, GMT-0 GMT0, Greenwich, HST, Hongkong Iceland, Indian/Antananarivo, Indian/Chagos, Indian/Christmas Indian/Cocos, Indian/Comoro, Indian/Kerguelen, Indian/Mahe Indian/Maldives, Indian/Mauritius, Indian/Mayotte, Indian/Reunion Iran, Israel, Jamaica, Japan Kwajalein, Libya, MET, MST MST7MDT, Mexico/BajaNorte, Mexico/BajaSur, Mexico/General NZ, NZ-CHAT, Navajo, PRC PST8PDT, Pacific/Apia, Pacific/Auckland, Pacific/Bougainville Pacific/Chatham, Pacific/Chuuk, Pacific/Easter, Pacific/Efate Pacific/Enderbury, Pacific/Fakaofo, Pacific/Fiji, Pacific/Funafuti Pacific/Galapagos, Pacific/Gambier, Pacific/Guadalcanal, Pacific/Guam Pacific/Honolulu, Pacific/Johnston, Pacific/Kiritimati, Pacific/Kosrae Pacific/Kwajalein, Pacific/Majuro, Pacific/Marguesas, Pacific/Midway Pacific/Nauru, Pacific/Niue, Pacific/Norfolk, Pacific/Nou Pacific/Pago Pago, Pacific/Palau, Pacific/Pitcairn, Pacific/Pohnpei Pacific/Ponape, Pacific/Port_Moresby, Pacific/Rarotonga, Pacific/Saipan Pacific/Samoa, Pacific/Tahiti, Pacific/Tarawa, Pacific/Tongatapu Pacific/Truk, Pacific/Wake, Pacific/Wallis, Pacific/Yap

Poland, Portugal, ROC, ROK Singapore, Turkey, UCT, US/Alaska US/Aleutian, US/Arizona, US/Central, US/East-Indiana US/Eastern, US/Hawaii, US/Indiana-Starke, US/Michigan US/Mountain, US/Pacific, US/Samoa, UTC Universal, W-SU, WET, Zulu

4.6. Alerts

4.6.1. Overview

Log Service provides the alerting feature. You can configure alert rules to trigger alerts based on query and analysis results. After you create an alert rule, Log Service checks related query and analysis results on a regular basis. If a query and analysis result meets the trigger condition that you specify in the alert rule, Log Service sends an alert notification. This way, you can monitor the service status in real time.

Limits

The following table describes the limits of the alerting feature in Log Service.

Item	Description
Associated query statements	You can associate an alert rule with a maximum of three query statements.
Field value size	If the number of characters that are included in a field exceeds 1,024, Log Service extracts only the first 1,024 characters for data processing.
Trigger condition	 Trigger conditions have the following limits: Each trigger condition must be 1 to 128 characters in length. If a query result includes more than 100 rows, Log Service only checks whether the first 100 rows meet the specified trigger condition. Log Service checks whether a trigger condition is met for a maximum of 1,000 times for the specified query statements.
Query time range	The maximum time range that you can specify for each query is 24 hours.
Voice calls	If a voice call is not answered, Log Service sends an SMS notification. You are charged only once for the voice call regardless of whether the call is answered. You are not charged for SMS notifications.

Query statements in alert rules

An alert rule is associated with one or more charts in a dashboard. Each chart displays the result of a query statement. You can associate an alert rule with one or more search statements or query statements.

• A search statement returns the log entries that meet the specified search condition.

For example, you can execute the **error** statement to search for the log entries that are generated in the previous 15 minutes and contain**error**. A total of 154 log entries are returned. Each log entry consists of key-value pairs. You can specify a trigger condition based on the value of a key.

Note If the number of returned log entries exceeds 100, Log Service checks only the first 100 log entries. If one of the log entries meets the specified condition, an alert is triggered.

🗟 internal-diagno:	stic_log				OApr 3, 2019, 11:39:30 ~ Apr 3, 2019	0, 11:40:00 V Share	Index Attributes	Save Search Saved as Alarm
1 error							© 🕜 🕓	earch & Analysis
0	1	Start Time: J End Time: A Occurrence: The search r	Apr 3, 2019, 11:39:38 pr 3, 2019, 11:39:39 s: 0 results are accurate.	11:39:39 11:39:42 11:39:45	11:39:48 11:39:51	11:39:54	11:39:5	7
				Log Entries:5 Search Status:The results are ac	curate.			
Raw Logs	LogRe	educe new	LiveTail Grap	h		Display Content Colum	n Column Set	ttings [🖓
Quick Analysis		<	Time 🔺	Content				
alarm_count	۲	1	Apr 3, 11:39:59	source: log_service topic: logtail_status				
alarm_type	۲			cpu: 0.006332278 v detail_metric: {}				
begin_time	۲			config_get_last_time: "2019-04-03 03:39:47" config_pet_last_time: "false"				
config_name	۲			config_update_count: "2" config_update_item_count: "2"				
consumer_group	config update_last_time: "2019-03-27 16:15:28" env_config: True" env_config: True"							
сри	۲			event_tps: "2.825" last_read_event_time: "2019-04-03 03:39:36"				
detail_metric				last send time: "2019-04-03 03:39:36" multi_config: "false"				

• A query statement consists of a search statement and an analytic statement. The analytic statement analyzes the log entries that meet the search condition and returns a result.

For example, the * | select sum(case when status='ok' then 1 else 0 end) *1.0/count(1) as ratio statement returns the percentage of the log entries in which the value of the status field is ok. If you set the trigger condition of an alert rule to ratio < 0.9, an alert is triggered if the percentage of the log entries whose status code is ok is less than 90%.

Cloud Defined Storage

🗟 interna	l-diagn	ostic_log														© 1	.5Minu	tes(Relative) 🔻	Share	Index	Attributes	Save Search	Saved as Alarm
1 * sele	ect sum(o	ase when	status='o	ok' then	1 else 0) end) *	1.0/count	(1) as rati	0													0 🖗	Search & Analysis
6			_	_																	_		
	_		_																	_			
11:37:51	1	11	39:15		11:40	0:45	l	11:42:15		11:43:4	5	1	1:45	:15	11	:46:45		11:48:15		11:49:4	5	11:51:15	11:52:86
								Log Ent	ries: 81 Sea	arch Status	The resu	lts are a	iccu	rate. Scanne	ed Rows	: 81 Sea	arch Tin	ne:210ms					
Raw Log	IS	LogF	leduce 🚾		LiveTa	ail	Gr	aph															
FR	~	lil.	=	•		~	123	-	222		54		۲	J 💰	Þ	<	-8	.mmp		7	bbe .		
Chart Prev	iew							Add to	New Dashl	board	Download	i Log		Data Source	Pr	operties	s Ir	nteractive Beha	vior				Hide
ratio												*	ç)uery:									
0.0040045	0700400	10												* select sun	n(case w	vhen sta	itus='ol	k' then 1 else 0	end) *1.0/c	ount(1) as	ratio		
0.3012343073012340								~	Select the query statement to generate a placeholder variable. You can configure a drill-down configuration tr						guration to replace								
													ti F	he variable. or how to use	e dashb	oards, p	olease n	efer to the doci	umentation	(Help)			

4.6.2. Configure an alarm

4.6.2.1. Configure an alert rule

You can create an alert rule on the Search & Analysis page of a Logstore or on a dashboard page in the Log Service console. After you create an alert rule, Log Service sends alert notifications when the trigger condition in the alert rule is met. This topic describes how to create an alert rule in the Log Service console.

Prerequisites

- Logs are collected and stored in a Logstore.
- The indexing feature is enabled and indexes are configured. For more information, see Configure indexes.

Background information

Log Service allows you to configure alert rules based on charts. You can create an alert rule for a query statement on the Search & Analysis page. After you create the alert rule, a chart that shows the query result of the query statement is automatically created on the specified dashboard. You can also create an alert rule for one or more existing charts on a dashboard.

- · Create a chart and configure an alert rule for the chart
- After you create an alert rule for a query statement, a chart that shows the query result of the query statement is automatically created on the specified dashboard. When you create an alert rule for a query statement on the Search & Analysis page, you must specify a dashboard and a name for the chart.
- Create an alert rule for existing charts on a dashboard

If you create an alert rule for multiple existing charts, you can specify one or more charts with which you want to associate the alert rule. You can specify a conditional expression for each chart, and combine the conditional expressions into a trigger condition of the alert rule.

The following section describes how to configure an alert rule for multiple existing charts on a dashboard.

⑦ Note If you modify the query statement of a chart with which an alert rule is associated, you must update the query statement in the alert rule. For more information, see Modify an alert rule.

For information about common configurations for alert rules, see FAQ about alerts.

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. In the left-side navigation pane, click the ricon.
- 4. In the Dashboard list, click the dashboard that you want to manage.
- 5. Click i on the specify chart, choose **Create Alerts**.
- 6. In the Alert Configuration step, configure the parameters and click Next.

The following table describes the parameters.

Parameter	Description
Alert Name	The name of the alert rule. The name must be 1 to 64 characters in length.
Associated Chart	The chart with which you want to associate the alert rule. You can add up to three charts. You can configure an alert rule for up to three query statements at the same time. The number before the chart name is the serial number of the chart. The serial number of the chart is valid in the alert rule. You can use the serial number to specify a chart in the Trigger Condition parameter. You can click the <i>i</i> icon next to the Query field to modify the query statement. The Search Period parameter specifies the time range of each query. You can select a relative time or a time frame. For example, the current time is 14:30:06 . • If you set the Search Period parameter to 15 Minutes(Relative) , the time range of the query is 14:15:06-14:30:06 .

Frequency	The frequency at which query results are checked.
Trigger Condition	The trigger condition of an alert. If the specified trigger condition is met, an alert is triggered and alert notifications are sent based on the values of the Frequency and Notification Interval parameters. For example, you can set the trigger condition to pv%100 > 0 && uv > 0. (?) Note You can use \$serial number to specify a chart in a trigger condition. For example, \$0 specifies chart 0.
Advanced	
Notification Trigger Threshold	An alert is triggered only if the specified trigger condition is met during continuous check periods. If the number of continuous triggers reaches the specified threshold, alert notifications are sent at the specified notification interval. If the trigger condition is not met, no alert is triggered. Default value: 1. This value specifies that alert notifications are sent if the trigger condition is met. You can set the Notification Trigger Threshold parameter to an integer that is greater than 1. In this case, alert notifications are sent only if the number of continuous triggers reaches the threshold. For example, you set the Notification Trigger Threshold parameter to 100. In this case, if the trigger condition is met for 100 times during continuous check periods, the value of Notification Trigger Threshold is reached. If the interval between the current time and the last time when alert notifications are sent exceeds the specified value of the Notification Interval parameter, an alert notification is sent. After an alert notification is sent, Log Service resets the number of continuous triggers to zero. If a check fails due to network exceptions, the check is not counted.
Notification Interval	The interval at which Log Service sends alert notifications. If the trigger condition is met in a check, Log Service checks whether the number of continuous triggers reaches the specified value of the Notification Trigger Threshold parameter. Log Service also checks whether the interval between the current time and the last time when alert notifications are sent exceeds the specified value of the Notification Interval parameter. If you set the Notification Interval parameter to 5 minutes, only one alert notification is received once every 5 minutes. Note You can use the Notification Trigger Threshold and Notification Interval parameters to control the number of alert notifications that you receive.

7. In the Notifications step, configure alert notification methods and click Submit.

Log Service supports the following alert notification methods: WebHook-Custom and WebHook-DingTalk Bot. For more information, see Configure

4.6.2.2. Authorize a RAM user to manage alert rules

You can use your Apsara Stack tenant account to authorize a RAM user to manage alert rules. This topic describes how to create a RAM user and authorize the RAM user to manage alert rules.

Procedure

- 1. Log on to the ASCM console ASCM console as an administrator.
- 2. Create a RAM role.

3. Create a permission policy. Replace the content in the Policy Document field with the following script. Replace *Project name* in the script with the name of your Log Service project.

t i i i i i i i i i i i i i i i i i i i
"Version": "1",
"Statement": [
{
"Effect": "Allow",
"Action": [
"log:CreateLogStore",
"log:CreateIndex",
"log:UpdateIndex"
],
"Resource": "acs:log:*:*:project/Project name/logstore/internal-alert-history"
},
{
"Effect": "Allow",
"Action": [
"log:CreateDashboard",
"log:CreateChart",
"log:UpdateDashboard"
],
"Resource": "acs:log:*:*:project/Project name/dashboard/*"
},
{
"Effect": "Allow",
"Action": [
"log:*"
],
"Resource": "acs:log:*:*:project/Project name/job/*"
}
]
}

4. Create a user.

- 5. Create a RAM user group.
- 6. Add a RAM user to a RAM user group
- 7. Grant permissions to a RAM role.

4.6.2.3. Configure alert notification methods

Log Service supports the following alert notification methods: Webhook-Custom and WebHook-DingTalk Bot. This topic describes how to configure alert notification methods.

Webhook-Custom

If you set the notification method to WebHook-Custom, Log Service sends alert notifications to a custom webhook URL.

(2) Note If Log Service does not receive a response within 5 seconds after a notification is sent, the request times out.

 When you configure an alert rule, select WebHook-Custom from the Notifications drop-down list. For more information, see Configure an alert rule.

2. Configure the parameters. The following table describes the parameters.

Parameter	Description
Request URL	The custom webhook URL.
Request Method	The method that is used to send the notification. The following request methods are supported: GET, POST, DELETE, PUT, and OPTIONS. The default request header is Content-Type: application/json;charset=utf-8 . To add request headers, click Add Request Headers .
Request Content	The content of the alert notification. Log Service provides default content. The content must be 1 to 500 characters in length. You can specify custom content. You can also use template variables in the content. For more information, see Template variables.

3. Click Submit.

WebHook-DingTalk Bot

If you set the notification method to Webhook-DingTalk Bot, Log Service sends alert notifications to the DingTalk group to which a specified webhook URL points by using a DingTalk chatbot. The chatbot can also remind the specified contacts of the alert notifications.

Note Each DingTalk chatbot can send up to 20 alert notifications per minute.

1. Create a DingTalk chatbot.

- i. Open DingTalk and go to a DingTalk group.
- ii. In the upper-right corner of the chat window, click the Group Settings icon and choose Group Assistant > Add Robot.
- iii. In the ChatBot dialog box, click the + icon in the Add Robot section.
- iv. In the Robot details dialog box, select Custom (Custom message services via Webhook) and click Add.
- v. In the Add Robot dialog box, enter a chatbot name in the Chatbot name field and select security options in the Security Settings section based on your business requirements. Then, select I have read and accepted DingTalk Custom Robot Service Terms of Service and click Finished.

O Note We recommend that you set the Security Settings parameter to Custom Keywords. You can specify up to 10 keywords. The chatbot sends only messages that contain at least one of the specified keywords. We recommend that you specify Alert as a keyword.

vi. Click **Copy** to copy the webhook URL.

2. Configure a notification method in the Log Service console.

- i. When you configure an alert rule, select **WebHook-DingTalk Bot** from the **Notifications** drop-down list. For more information, see Configure an alert rule.
- ii. Configure the parameters. The following table describes the parameters.

Parameter	Description						
Request URL	The webhook URL of the DingTalk chatbot. Paste the webhook URL that you copied in Step 1.						
Title	The alert topic. The title must be 1 to 100 characters in length. You can specify a custom title. You can also use template variables in the title. For more information, see Template variables.						
Recipients	The group members whom you want to remind of the alert notification. Valid values: None, All, and Specified Members. If you select Specified Members , enter the mobile phone numbers of the group members in the Tagged List field. Separate multiple mobile phone numbers with commas (,).						
Content	The content of the alert notification. Log Service provides default content. You can modify the content based on your business scenario. The content must be 1 to 500 characters in length. You can specify custom content. You can also use template variables in the content. For more information, see Template variables.						

iii. Click Submit.

Template variables

You can use template variables when you configure a notification method for an alert rule. When you configure the **Content** and **Subject** parameters, you can use the *\${fieldName}* syntax to reference a template variable. When Log Service sends an alert notification, Log Service replaces the template variables that are referenced in the **Content** and **Subject** parameters with actual values. For example, Log Service replaces *\${Project}* with the name of the project to which the alert rule belongs.

() Important You must reference valid variables. If a referenced variable does not exist or is invalid, Log Service processes the variable as an empty string. If the value of a referenced variable is of the object type, the value is converted and displayed as a JSON string.

The following table describes the supported template variables and the methods that are used to reference the variables.

Variable	Description	Example	Reference example
Aliuid	The ID of the Apsara Stack tenant account to which the project belongs.	1234567890	An alert is triggered for the Apsara Stack tenant account \${Aliuid}.
Project	The project to which an alert rule belongs.	my-project	An alert is triggered in the \${Project} project.
AlertID	The ID of an alert.	0fdd88063a611aa114938f9371daeeb6- 1671a52eb23	The ID of the alert is \${AlertID}.
AlertName	The ID of an alert rule. The ID is unique in a project.	alert-1542111415-153472	An alert is triggered based on the \${AlertName} alert rule.
AlertDisplayName	The display name of an alert rule.	My alert	An alert is triggered based on the \${AlertDisplayName} alert rule.
Condition	The conditional expression that triggers an alert. In an alert notification, a variable is replaced by an actual value that is enclosed in brackets [].	[5] > 1	The conditional expression that triggers an alert is \${Condition}.
RawCondition	The original conditional expression that triggers an alert.	count > 1	The original conditional expression that triggers an alert is \${RawCondition}.
Dashboard	The name of the dashboard that is associated with an alert rule.	mydashboard	The alert rule is associated with the \${Dashboard} dashboard.
DashboardUrl	The URL of the dashboard that is associated with an alert rule.	https://sls.console.aliyun.com/next/project/ myproject/dashboard/mydashboard	The URL of the dashboard that is associated with the alert rule is \${DashboardUrl}.
FireTime	The time when an alert is triggered.	2018-01-02 15:04:05	The alert is triggered at \${FireTime}.
FullResultUrl	The URL that is used to query the details of an alert.	https://sls.console.aliyun.com/next/project/ my-project/logsearch/internal-alert-history? endTime=1544083998&queryString=AlertI D%3A9155ea1ec10167985519fccede4d5fc 7- 1678293caad&queryTimeType=99&startTi me=1544083968	Click \${FullResultUrl} to view the alert details.

Cloud Defined Storage

Results	The parameters and results of a query. The value is of the array type. For more information, see Fields in alert logs. ⑦ Note The Results variable can contain the information of up to 100 alerts.	<pre>[{ "EndTime": 1542507580, "FireResult": { "_time": "1542453580", "count": "0" }, "LogStore": "test-logstore", "Query": "* SELECT COUNT(*) as count", "RawResultCount": 1, "RawResultS: [{</pre>	The first query starts at \${Results[0].5tartTime} and ends at \${Results[0].EndTime}. The alert is triggered \${Results[0].FireResult.count} times. Image: The second start of the second start
---------	---	--	---

4.6.3. Modify and view an alarm

4.6.3.1. Modify an alert rule

This topic describes how to modify an alert rule. After you create an alert rule, you can modify the chart that is associated with the alert rule, and then update the alert rule. If you associate a query statement with an alert rule, you can modify the query statement to modify the alert rule.

Procedure

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. In the left-side navigation pane, click the real icon.
- 4. In the Alerts list, click the alert rule that you want to modify.
- 5. On the Alert Overview page, click Modify Settings.
- In the Modify Alert panel, modify the chart that is associated with the alert rule. In the Associated Chart section, select the chart that you want to associate with the alert rule.
- 7. Modify the query statement that is associated with the alert rule.
 - i. In the Associated Chart section, find the query statement that you want to modify. Then, click the ricon next to the Query field.

O Note If the original statement consists of only a search statement, the new statement must also consist of only a search statement. If the original statement consists of a search statement and an analytic statement, the new statement must also consist of a search statement and an analytic statement. For example, after you associate the search statement request_method: GET with an alert rule, you can change the search statement to error! select count(1) as c.

- ii. In the Edit dialog box, enter a new query statement and click Preview.
- iii. If the expected query result is returned, click OK.
- Modify the evaluation frequency, trigger condition, notification threshold, and notification interval. Then, click Next. For more information, see Configure an alert rule.
- Modify the alert notification method and click Submit. For more information, see Configure alert notification methods.

4.6.3.2. View alert statistics

Log Service records historical alert statistics in logs and automatically creates a dashboard to display the execution results of all alert monitoring rules and the alert notifications that are sent.

Background information

• View alert logs in a Logstore After you create an alert monitoring rule, Log Service automatically creates a Logstore named **internal-alert-history** in the project to which the alert monitoring rule belongs. A log is generated and written to the Logstore each time an alert monitoring rule is executed in the project, regardless of whether an alert is triggered. For more information about log fields, see Fields in alert logs.



View alert statistics in a dashboard

After you create an alert monitoring rule, Log Service automatically creates a dashboard named **Alert History Statistics** in the project to which the alert monitoring rule belongs. The dashboard displays information about all alerts that are triggered in the current project. The information includes Alerts, Execution Success Rate, Notification Rate upon Successful Execution, and Top 10 Alert Rule Executions.

(2) Note The Alert History Statistics dashboard cannot be deleted or modified.

View alert logs in a Logstore

1. Log on to the Simple Log Service console

2. In the Projects section, click the required project,

On the Log Storage > Logstores tab, find the internal-alert-history Logstore and choose || > Search and Analysis.

4. On the page that appears, query alert logs based on your business requirements.

View the Alert History Statistics dashboard

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the required project.
- 3. In the left-side navigation pane, click the ricon.

4. In the Dashboard list, click **Alert History Statistics**.

The **Alert History Statistics** dashboard displays information about alerts, such as whether an alert is triggered, the reason why the alert is triggered, and the error information and description of the alert.



4.6.3.3. Manage alerts

After you configure an alert, you can manage the alert. For example, you can view information about the alert, modify the alert, and delete the alert.

View information about an alert

- 1. Log on to the Simple Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. In the left-side navigation pane, click the Alerts icon.
- 4. In the list of alerts, click the alert whose information you want to view to go to the Alert Overview page.
- 5. On the Alert Overview page, view the information about the alert. On the Alert Overview page, you can view the dashboard to which the alert belongs, the point in time when the alert is configured, the most recent point in time when the alert is updated, the check frequency of the alert, the status of the alert, and the status of the alert notification feature for the alert.

I Basic Information					
Dashboard	line and	Created At	May 10, 2023, 13:57:46		
Last Updated At	May 10, 2023, 13:57:46	Check Frequency	Fixed Interval 15Minutes		
Status	Enabled	Monitoring Status	Enabled	Modify	

Disable or enable an alert

After you configure an alert, you can disable or enable the alert at any time.

③ Note After an alert is disabled, Log Service no longer checks data for the alert and no longer sends alert notifications for the alert.

On the Alert Overview page, click Disabled or Enabled on the right side of Status.

Basic Informatio	n			
Dashboard	1010	Created At	May 10, 2023, 13:57:46	
Last Updated At	May 10, 2023, 13:57:46	Check Frequency	Fixed Interval 15Minutes	
Status	Enabled	Monitoring Status	Enabled	Modify

Suspend or resume the alert notification feature for an alert

If the monitoring status of an alert is **Enabled**, you can suspend the alert notification feature for the alert.

③ Note If the alert notification feature is suspended for an alert, Log Service regularly checks data for the alert. However, Log Service does not send alert notifications even if the alert trigger condition is met.

1. On the Alert Overview page of an alert, click Modify on the right side of Monitoring Status.

2. Configure Disabled Duration and click **OK**.

After the alert notification feature is suspended, you can view the value of **Monitoring Status** to determine the point in time when the alert notification feature is scheduled to resume. If you want to resume the alert notification feature before the scheduled point in time, you can click **Modify** on the right side of **Monitoring Status**. In the message that appears, click OK.

Basic Information					
Dashboard	eee		Created At	Apr 14, 2020, 16:50:53	
Last Updated At	Apr 14, 2020, 16:50:53		Check Frequency	Fixed Interval 15Minutes	
Status	Enabled	Close	Notification Status	Enabled	Modify
	la ut				

Delete an alert

▲ Warning After you delete an alert, the alert cannot be restored. Proceed with caution.

In the upper-right corner of the Alert Overview page, click Delete Alert.

4.6.4. Relevant syntax and fields for reference

4.6.4.1. Syntax of conditional expressions in alert rules

Log Service checks whether the specified trigger conditions are met based on the execution results of conditional expressions specified in alert rules. The result of a specified query statement is used as input, and the fields in the result of set operations are used as variables. If the condition that is specified in a conditional expression is met, an alert is triggered.

Limits

The conditional expressions that you can specify in an alert rule have the following limits:

- Negative numbers must be enclosed in parentheses (), for example, x+(-100)<100.
- Numeric values are converted to 64-bit floating-point numbers. If a comparison operator such as equal-to (==) is used, errors may occur.
- Variable names can contain only letters and digits, and must start with a letter.
- A conditional expression must be 1 to 128 characters in length.
- You can specify up to 1,000 conditions in a conditional expression. If the evaluation result of each condition in a conditional expression is false, the evaluation result of the conditional expression is false.
- An alert rule can be associated with a maximum of three charts or query statements.
- An alert is triggered only if the result of the specified conditional expression is true. For example, if a conditional expression is 100+100, the result is 200 and is not true, and no alert is triggered.
- Log Service reserves the words true and false. Log Service also reserves the special characters dollar sign (\$) and period (.). You cannot use the reserved words and special characters as variables.

Syntax

The following table describes the types of syntax that is supported for the conditional expressions of an alert rule.

Syntax type	Description	Example
Arithmetic operators	The addition (+), subtraction (-), multiplication (*), division (/), and modulus (%) operators are supported. +-*/%	 x * 100 + y > 200 x % 10 > 5
Comparison operators	The following eight comparison operators are supported: greater-than (>), greater-than-or-equal-to (>=), less-than (<), less-than-or-equal-to (<=), equal-to (==), not-equal-to (!=), regex match (=~), and regex not match (!~). Note • Backslashes (\) in regular expressions must be escaped. • The Regular Expression 2 (RE2) syntax is used.	 x >= 0 x < 100 x <= 100 x == "foo" Regex match: x =~ "\\w +"
Logical operators	The AND (&&) and OR () operators are supported.	 x >= 0 && y <= 100 x > 0 y >0
Not operator	The not operator (!) is supported.	!(a < 1 && a > 100)
Numeric constants	Numeric constants are supported. Log Service converts numeric constants to 64- bit floating-point numbers.	x > 100
String constants	String constants are supported. The string constants are in the format of String ', for example, 'string'.	foo == 'string'
Boolean constants	Boolean constants are supported. Valid values: true and false.	(x > 100) == true
Parentheses	Parentheses () can be used to override the standard precedence order and force Log Service to evaluate the enclosed part of a trigger condition before an unenclosed part.	x * (y + 100) > 100
contains function	The contains function can be used to check whether a string contains a substring. For example, if you invoke contains(foo,'hello') and true is returned, this indicates that the foo string contains the hello substring.	<pre>contains(foo, 'hello')</pre>

Evaluate the results of multiple query statements

• Syntax

An alert rule can be associated with multiple query statements. If you want to use a variable in the trigger condition to reference a field from the result of a query statement, you must prefix the variable with the serial number of the query statement in the **\$N.fieldname** format. The serial number of a query statement is the same as the serial number of the chart that shows the result of the query statement. Each alert rule can be associated with up to three query statements. Therefore, the value range of N is 0 to 2. For example, **\$0.foo** references the value of the **foo** field from the result of the first query statement. If an alert rule is associated with only one query statement, you do not need to specify the prefix in the trigger condition.

Evaluate a conditional expression

If multiple query results are returned, the variables specified in the conditional expression specify the results that need to be used to evaluate the conditional expression. For example, three query statements are specified and three sets of query results are returned. The number of log entries in the first set is x, the number of log entries in the second set is y, and the number of log entries in the third set is z. If the conditional expression that you specify is 0.660 > 100 & 1.640, only the first two sets are used to evaluate the conditional expression. Up to x × y times of evaluation or 1,000 if x × y is greater than 1,000, is performed. If the conditional expression is met within the maximum number of times of evaluation, true is returned. Otherwise, false is returned.

Operation methods

? Note

- Log Service converts all numeric values to 64-bit floating-point numbers.
- A string constant must be enclosed in single quotation marks (") or double quotation marks (""), for example, 'String' or "String".
- Boolean values include true and false.

	Operation method					
Operator	Operation between variables	Operation between a non-string constant and a variable	Operation between a string constant and a variable			
Arithmetic operators: addition (+), subtraction (-), multiplication (*), division (/), and modulus (%)	Before an arithmetic operator is applied, the left and right operands are converted to 64-bit floating-point numbers.	Before an arithmetic operator is applied, the left and right operands are converted to 64-bit floating-point numbers.	Not supported.			
Comparison operators: greater-than (>), greater-than-or-equal-to (>=), less-than (<), less- than-or-equal-to (<=), equal-to (==), and not- equal-to (!=)	Log Service uses the following comparison rules that are sorted in the precedence order:1. The left and right operands are converted to 64-bit floating-point numbers, and then compared based on the numerical order. If the conversion fails, the operation of the next priority is performed.2. The left and right operands are converted to strings, and then compared based on the lexicographic order.	The left and right operands are converted to 64-bit floating-point numbers, and then compared based on the numerical order.	The left and right operands are converted to strings, and then compared based on the lexicographic order.			
Matching operators: regex match (= \sim) and regex not match (! \sim)	Before a matching operator is applied, the left and right operands are converted to strings.	Not supported.	Before a matching operator is applied, the left and right operands are converted to strings.			
Logical operators: AND (&&) and OR ()	A logical operator cannot be applied to log fields. The left and right operands must be sub-expressions and the result of the operation must be a Boolean value.	A logical operator cannot be applied to log fields. The left and right operands must be sub- expressions and the result of the operation must be a Boolean value.	A logical operator cannot be applied to log fields. The left and right operands must be sub- expressions and the result of the operation must be a Boolean value.			
Not operator (!)	The not operator cannot be applied to log fields. The specified operand must be a sub-expression and the result of the operation must be a Boolean value.	The not operator cannot be applied to log fields. The specified operand must be a sub- expression and the result of the operation must be a Boolean value.	The not operator cannot be applied to log fields. The specified operand must be a sub- expression and the result of the operation must be a Boolean value.			
contains function	Before the contains function is run, the left and right operands are converted to strings.	Not supported.	Before the contains function is run, the left and right operands are converted to strings.			
Parentheses ()	Parentheses () are used to override the standard precedence order and force Log Service to evaluate the enclosed part of a trigger condition before an unenclosed part.	Parentheses () are used to override the standard precedence order and force Log Service to evaluate the enclosed part of a trigger condition before an unenclosed part.	Parentheses () are used to override the standard precedence order and force Log Service to evaluate the enclosed part of a trigger condition before an unenclosed part.			

4.6.4.2. Fields in alert logs

After you configure an alert rule, Log Service automatically creates a Logstore to store log entries that are generated when the alert rule is executed and alert notifications are sent. This topic describes the fields in alert logs.

Fields in the logs that are generated when an alert rule is executed

Log field	Description	Example
AlertDisplayName	The display name of an alert rule.	Test alert rule
AlertID	The ID of an alert. The ID is unique.	0fdd88063a611aa114938f9371daeeb6-1671a52eb23

User Guide-Log Service

Cloud Defined Storage

AlertName	The name of the alert rule. The name is unique within a project.	alert-1542111415-153472
Condition	The conditional expression of an alert rule.	\$0.count > 1
Dashboard	The dashboard that is associated with the alert rule.	my-dashboard
FireCount	The cumulative number of triggers since the last alert notification was sent.	1
Fired	Indicates whether an alert was triggered. Valid values: true and false.	true
LastNotifiedAt	The time when the last alert notification was sent. The value is a UNIX timestamp.	1542164541
NotifyStatus	 The notification status of an alert. Valid values: Success: Alert notifications were sent. Failed: Alert notifications failed to be sent. NotNotified: No alert notification was sent. PartialSuccess: Some of the alert notifications were sent. 	Success
Reason	The reason why alert notifications failed to be sent or no alert notification was sent.	result type is not bool
Results	The parameters and results of each log query. The value is of the array type. For more information, see Results field.	<pre>[</pre>
Status	The execution result of an alert. Valid values: Success and Failed.	Success

Results field

Log field	Description	Example
Query	The query statement.	* select count(1) as count
LogStore	The Logstore in which data is queried.	my-logstore
StartTime	The beginning of the time range for a query.	2019-01-02 15:04:05
StartTimeTs	The beginning of the time range for a query. The value is a UNIX timestamp.	1542334840
EndTime	The end of the time range for a query.	2019-01-02 15:19:05
EndTimeTs	The end of the time range for a query. The value is a UNIX timestamp. The actual query time range is [StartTime, EndTime] .	1542334900
RawResults	The query result that is formatted in an array. Each element in the array is a log entry. An array can contain a maximum of 100 elements.	[{ "time": "1542334840", "count": "0" }
RawResultsAsKv	The query result that is formatted in key-value pairs. ⑦ Note This field can be used as a template variable. No data is stored to a Logstore for this field.	[foo:0]
RawResultCount	The number of raw log entries that are returned.	1

FireResult	The log entry that records the triggers of an alert. If no alert is triggered, the value is NULL.	{ "time": "1542334840", "count": "0" }
	The log entry that records the triggers of an alert. The log entry is formatted in key-value pairs.	
FireResultAsKv	⑦ Note This field can be used as a template variable. No data is stored to a Logstore for this field.	[foo:0]

4.6.5. FAQ

4.6.5.1. A DingTalk alert notification fails to be sent and the error message "send webhook failed: unable to send request: Post" is reported. Why?

Issue

A DingTalk alert notification fails to be sent and the error message send webhook failed: unable to send request: Post **** is reported.

Cause

The DingTalk certificate is not loaded on the physical machine on which Simple Log Service is deployed.

Solution

Download the DingTalk certificate, import the certificate to the physical machine on which Simple Log Service is deployed, and then restart the SIsEtIFramework# server role.

1. Download the DingTalk certificate.

i. Use a browser to visit https://oapi.dingtalk.com .

The following information is returned:

{"errcode":404,"errmsg":"The requested URI does not exist."}

- ii. Open the browser developer tool, click Security, and then view the certificate.
- iii. On the details tab of the View certificate dialog box, click Copy to File. Follow the wizard to save the DER-encoded binary certificate file to your computer and name the file oapi.dingtalk.com-cert.pem.cer.
- Upload the DingTalk certificate to the physical machine on which Simple Log Service is deployed and add the certificate to the trusted certificate list.
 i. Log on to the physical machine and upload the certificate to the physical machine.
 - ii. Run the following command to convert the format of the certificate file:

openssl x509 -inform der -in oapi.dingtalk.com-cert.pem.cer -out oapi.dingtalk.com-cert.pem

In this command, oapi.dingtalk.com-cert.pem.cer specifies the certificate file that you download and oapi.dingtalk.com-cert.pem specifies the converted certificate file based on the actual situation.

iii. Run the following command to obtain the path to the converted certificate file and the name of the certificate file that you need to back up:

curl -v https://oapi.dingtalk.com/

Int	formation similar to the following example is returned.
[a	dmin@a11a05112.uluum.ump.l.put /home/admin]
Şc	url -v https://oapi.dingtalk.com/
*	About to connect() to oapi.dingtalk.com port 443 (#0)
*	Trying 172
×	Connected to capi.dingtalk.com (17 . 2 . 3 . 3) port 443 (#0)
*	Initializing NSS with certpath: sql:/etc/pki/nssdb
*	CAfile: /etc/pki/tls/certs/ca-bundle.crt
	CApath: none
*	Server certificate:
*	subject: CN=oapi.dingtalk.com,O=Default Company Ltd,L=Default
Ci	ty,C=XX
*	start date: Jul 13 04:44:03 2020 GMT
*	expire date: Jul 11 04:44:03 2030 GMT
*	common name: oapi.dingtalk.com
*	issuer: CN=oapi.dingtalk.com,O=Default Company Ltd,L=Default
Ci	ty,C=XX
*	NSS error -8172 (SEC_ERROR_UNTRUSTED_ISSUER)
*	Peer's certificate issuer has been marked as not trusted by the user

iv. Run the following command to back up the original certificate:

cp -aL /etc/pki/tls/certs/ca-bundle.crt ./ca-bundle.crt.bak

v. Run the following command to add the converted certificate to the trusted certificate list:

cat oapi.dingtalk.com-cert.pem >> /etc/pki/tls/certs/ca-bundle.crt

3. Restart the SIsEtlFramework# server role.

- i. Log on to the Apsara Uni-manager Operations Console.
- ii. In the top navigation bar of the Apsara Uni-manager Operations Console, click Products. Then, choose Base/Platforms > Apsara
- Infrastructure Management Console
- iii. In the left-side navigation pane, click **Product & Cluster O&M**. On the page that appears, enter **sign** in the search box. In the search result, click **Cluster Details** in the Actions column.
- iv. On the Cluster Details page, enter sls in the Service field. In the search result, click sls-backend-server.
- v. In the Components section, enter SIsEtIFramework# in the search box and click the component to open the component information page.
- vi. On the **Instances** tab, choose **Actions** > **Restart** in the Actions column for each instance.

4.7. Real-time consumption

4.7.1. Overview

Log Service provides the real-time log consumption feature that allows you to read and write full data in the first-in, first-out (FIFO) order. This feature is similar to the features provided by Kafka. This topic describes the types of real-time consumption.

The following table describes the methods that you can use to process log data after the log data is sent to Log Service.

Method	Scenario	Timeliness	Retention period
Real-time log consumption	Stream computing and real-time computing	Real time	You can specify a retention period based on your business requirements.
Log query	Online query of recent hot data	Near real time with a latency of no more than 3 seconds in all cases and a latency of no more than 1 second in 99.9% cases	You can specify a retention period based on your business requirements.
Log shipping	Storage of full log data for offline analysis	A latency of 5 to 30 minutes	The retention period is based on the storage system.

Real-time log consumption

Log Service allows you to pull log data and consume the data in real time. The following procedure describes how log data is consumed from a shard: 1. Obtain a cursor based on the start time and end time of data consumption.

- 2. Read log data based on the cursor and step parameters and return the position of the next cursor.
- 3. Move the cursor to continuously consume log data.
- Use an SDK to consume log data
- Log Service provides SDKs in multiple programming languages, such as Java, Python, and Go. You can use an SDK to consume log data.
- Use consumer groups to consume log data

Log Service provides an advanced method that allows you to consume log data by using consumer groups. A consumer group is a lightweight computing framework that allows multiple consumers to consume data from a Logstore at the same time. Shards are automatically allocated to consumers in a consumer group. Data is consumed in sequence based on the time when data is written to the Logstore. After a breakpoint, consumers can continue to consume data by using checkpoints. You can use consumer group SDKs in multiple programming languages, such as Go, Python, and Java, to consume log data.

- Use real-time computing systems to consume log data
 - Use Spark Streaming to consume log data.
 - Use Storm to consume log data
 - Use open source Flink to consume log data
- · Use open source services to consume log data

Use Flume to consume log data

4.7.2. Consume log data

This topic describes how to use an SDK to consume log data and preview log data in the Log Service console.

Use an SDK to consume log data

Log Service provides SDKs in various programming languages. You can use an SDK to consume log data. For more information, see the SDK Reference topic in **Log Service Developer Guide**. The following example shows how to use Log Service SDK for Java to consume log data from a shard:

Client client = new Client(host, accessId, accessKey);

```
String cursor = client.GetCursor(project, logStore, shardId, CursorMode.END).GetCursor();
System.out.println("cursor = " +cursor);
try {
    while (true) {
        PullLogsRequest request = new PullLogsRequest(project, logStore, shardId, 1000, cursor);
        PullLogsRequest request = new PullLogs(request);
        System.out.println(response.getCount());
        System.out.println("cursor = " + cursor + " next_cursor = " + response.getNextCursor());
        if (cursor.equals(response.getNextCursor())) {
            break;
            }
        cursor = response.getNextCursor();
        Thread.sleep(200);
    }
    }
    catch(LogException e) {
        System.out.println(e.GetRequestId() + e.GetErrorMessage());
    }
}
```

Preview log data in the Log Service console

Consumption preview is a type of log data consumption. The consumption preview feature allows you to preview specific log data that is stored in a Logstore in the Log Service console.

- 1. Log on to the Log Service console.
- 2. In the Projects section, click the project from which you want to consume log data.
- 3. In the Logstores list, find the Logstore from which you want to consume log data, click the 📓 icon next to the Logstore, and then select

Consumption Preview.

4. In the Consumption Preview panel, select a shard and a time range, and then click Preview. The Consumption Preview panel displays the log data of the first 10 packets in the specified time range.

Consumption Pre	eview)
internal-alert-histo	ory	Shard:0	V 15 Minutes N	Preview
Log preview is only used to check whether log data is uploaded successfully. If you want to search logs through keywords, enable log index.			search logs	
Time/Source	Content			
2020-05-07 10:40:10	C	","namespace":"nu	HII","today":"0"]],"StartTin	+a0 Su AI Fir Iiff d, bitr na LE Dje) a t :e es

4.7.3. Consumption by consumer groups

4.7.3.1. Use consumer groups to consume log data

If you use consumer groups to consume log data, you do not need to focus on factors such as Log Service implementation, load balancing among consumers, and failovers that may occur. This way, you can focus on the business logic during log data consumption.

Terms

Term	Description
consumer group	A consumer group consists of multiple consumers. Each consumer in a consumer group consumes different data in a Logstore. You can create a maximum of 30 consumer groups for a Logstore.
consumer	The consumers in a consumer group consume data. The name of each consumer in a consumer group must be unique.

A Logstore has multiple shards. A consumer library allocates shards to the consumers in a consumer group based on the following rules:

• You can allocate a shard to only one consumer.

• Each consumer can consume data from multiple shards.

After you add a consumer to a consumer group, shards that are allocated to the consumer group are reallocated to each consumer for load balancing. The shards are reallocated based on the preceding rules.

A consumer library stores checkpoints. This way, consumers can resume data consumption from a checkpoint and do not consume data after a program fault is resolved.

Procedure

```
You can use Java, Python, or Go to create consumers and consume data. The following procedure uses Java as an example:
1. Add Maven dependencies.
    <dependency>
      <groupId>com.google.protobuf</groupId>
      <artifactId>protobuf=java</artifactId>
      <version>2.5.0</version>
    </dependency>
    <dependency>
     <groupId>com.aliyun.openservices</groupId>
      <artifactId>loghub-client-lib</artifactId>
      <version>0.6.33</version>
    </dependency>
2. Create a file named Main.java.
    import com.aliyun.openservices.loghub.client.ClientWorker;
    import com.aliyun.openservices.loghub.client.config.LogHubConfig;
    import com.aliyun.openservices.loghub.client.exceptions.LogHubClientWorkerException;
    public class Main {
        \ensuremath{{\prime}}\xspace // The endpoint of Log Service. Set the parameter based on your business requirements.
        private static String sEndpoint = "cn-hangzhou.log.aliyuncs.com";
        \ensuremath{{\prime\prime}}\xspace // The name of a Log Service project. Set the parameter based on your business requirements.
        private static String sProject = "ali-cn-hangzhou-sls-admin";
        // The name of a Logstore. Set the parameter based on your business requirements.
        private static String sLogstore = "sls_operation_log";
        // The name of a consumer group. Set the parameter based on your business requirements.
private static String sConsumerGroup = "consumerGroupX";
        // The AccessKey pair that is used to access Log Service. Specify the AccessKey ID and AccessKey secret based on your business
    requirements.
        private static String sAccessKeyId = "";
        private static String sAccessKey = "";
        public static void main(String[] args) throws LogHubClientWorkerException, InterruptedException {
            // consumer_1 is the name of the consumer. The name of each consumer in a consumer group must be unique. If different consumers start
    multiple processes on multiple servers to consume the data of a Logstore, you can use a server IP address to identify a consume:
            The maxFetchLogGroupSize parameter specifies the maximum number of log groups that you can obtain from the server at the same time.
    Valid values: (0,1000]. We recommend that you use the default value.
            LogHubConfig config = new LogHubConfig(sConsumerGroup, "consumer_1", sEndpoint, sProject, sLogstore, sAccessKeyId, sAccessKey, LogHub
    Config.ConsumePosition.BEGIN_CURSOR);
            ClientWorker worker = new ClientWorker(new SampleLogHubProcessorFactory(), config);
            Thread thread = new Thread (worker);
            // After you execute the thread, the ClientWorker instance automatically runs and extends the Runnable interface.
            thread.start();
            Thread.sleep(60 * 60 * 1000);
            // The shutdown function of the ClientWorker instance is called to exit the consumption instance. The associated thread is
    automatically stopped.
           worker.shutdown();
            // Multiple asynchronous tasks are generated when the ClientWorker instance is running. To ensure that all running tasks exit after
    shutdown, we recommend that you set Thread.sleep to 30 seconds.
            Thread.sleep(30 * 1000);
    }
Create a file named SampleLogHubProcessor.iava.
```

import com.aliyun.openservices.log.common.FastLog; import com.aliyun.openservices.log.common.FastLogContent; import com.aliyun.openservices.log.common.FastLogGroup; import com.aliyun.openservices.log.common.FastLogTag; import com.aliyun.openservices.log.common.LogGroupData; import com.aliyun.openservices.loghub.client.ILogHubCheckPointTracker; import com.aliyun.openservices.loghub.client.exceptions.LogHubCheckPointException; import com.aliyun.openservices.loghub.client.interfaces.ILogHubProcessor; import com.aliyun.openservices.loghub.client.interfaces.ILogHubProcessorFactory; import java.util.List; public class SampleLogHubProcessor implements ILogHubProcessor { private int shardId; // The point in time when the last persistent checkpoint was saved. private long mLastCheckTime = 0; public void initialize(int shardId) { this.shardId = shardId; // The main logic of data consumption. You must include the code to handle all exceptions that may occur during log data consumption. public String process(List<LogGroupData> logGroups, ILogHubCheckPointTracker checkPointTracker) { // Display the data that you obtained. for (LogGroupData logGroup : logGroups) { FastLogGroup flg = logGroup.GetFastLogGroup(); System.out.println(String.format("\tcategory\t:\t%s\n\tsource\t:\t%s\n\ttopic\t:\t%s\n\tmachineUUID\t:\t%s", flg.getCategory(), flg.getSource(), flg.getTopic(), flg.getMachineUUID())); System.out.println("Tags"); for (int tagIdx = 0; tagIdx < flg.getLogTagsCount(); ++tagIdx) {</pre> FastLogTag logtag = flg.getLogTags(tagIdx); System.out.println(String.format("\t%s\t:\t%s", logtag.getKey(), logtag.getValue())); for (int lIdx = 0; lIdx < flg.getLogsCount(); ++lIdx) {</pre> FastLog log = flg.getLogs(lIdx); System.out.println("------\nLog: " + lIdx + ", time: " + log.getTime() + ", GetContentCount: " + log.getContentsCount()); for (int cIdx = 0; cIdx < log.getContentsCount(); ++cIdx) {</pre> FastLogContent content = log.getContents(cIdx); System.out.println(content.getKey() + "\t:\t" + content.getValue()); } long curTime = System.currentTimeMillis(); // Write a checkpoint to the server every 30 seconds. If the ClientWorker instance unexpectedly stops within 30 seconds, a newly star ted ClientWorker instance continues to consume data from the last checkpoint. A small amount of data may be repeatedly consumed. if (curTime - mLastCheckTime > 30 * 1000) { try { // If you set the parameter to true, checkpoints are immediately synchronized to the server. If you set the parameter to fals e, checkpoints are locally cached. The default value of a synchronization interval of checkpoints is 60 seconds. checkPointTracker.saveCheckPoint(true); } catch (LogHubCheckPointException e) { e.printStackTrace(); } mLastCheckTime = curTime; return null; } // The ClientWorker instance calls this function when the instance exits. You can delete the checkpoints. public void shutdown(ILogHubCheckPointTracker checkPointTracker) { // Save checkpoints to the server. try { checkPointTracker.saveCheckPoint(true); } catch (LogHubCheckPointException e) { e.printStackTrace(); } } } class SampleLogHubProcessorFactory implements ILogHubProcessorFactory { public ILogHubProcessor generatorProcessor() { // Generate a consumption instance. return new SampleLogHubProcessor(); } } For more information, see Java, Python, and Go.

View the status of a consumer group

You can use the Log Service console, call the API, or use an SDK to view the progress of your data consumption. For more information, see View the status of a consumer group.

Related operations

Handle exceptions.
 We recommend that you configure Log4j for the consumer program to return error messages in consumer groups. This way, you can handle

exceptions at the earliest opportunity. The following code shows a configuration file of log4j.properties:

log4j.rootLogger = info,stdout

- log4j.appender.stdout = org.apache.log4j.ConsoleAppender
- log4j.appender.stdout.Target = System.out
- log4j.appender.stdout.layout = org.apache.log4j.PatternLayout log4j_appender_stdout_layout_CompariseDattorn = [%=5n] %4(unuseDM-dd WW.mm.co. SSS) method.%1%p%m%n

log4j.appender.stdout.layout.ConversionPattern = [%-5p] %d{yyyy-MM-dd HH:mm:ss,SSS} method:%l%n%m%n

After you configure Log4j, you can view the information of exceptions that occur when you run the consumer program. The following example shows an error message:

[WARN] 2018-03-14 12:01:52,747 method:com.aliyun.openservices.loghub.client.LogHubConsumer.sampleLogError(LogHubConsumer.java:159) com.aliyun.openservices.log.exception.LogException: Invalid loggroup count, (0,1000]

• Use a consumer group to consume data that is generated from a certain point in time.

// consumerStartTimeInSeconds indicates that the data generated after the point in time is consumed.

// The value of the position parameter is an enumeration variable. LogHubConfig.ConsumePosition.BEGIN_CURSOR indicates that the consumption starts from the earliest data. LogHubConfig.ConsumePosition.END_CURSOR indicates that the consumption starts from the latest data. public LogHubConfig(String consumerGroupName,

String consumerName, String loghubEndPoint, String project, String logStore, String accessId, String accessKey, ConsumePosition position);

? Note

- You can use different constructors based on your business requirements.
- If a checkpoint is stored on the server, data consumption starts from this checkpoint.

Reset a checkpoint.

public static void updateCheckpoint() throws Exception {

- Client client = new Client(host, accessId, accessKey);
- long timestamp = Timestamp.valueOf("2017-11-15 00:00:00").getTime() / 1000;
- ListShardResponse response = client.ListShard(new ListShardRequest(project, logStore)); for (Shard shard : response.GetShards()) {
- int shardId = shard.GetShardId();
 - String cursor = client.GetCursor(project, logStore, shardId, timestamp).GetCursor();
- client.UpdateCheckPoint(project, logStore, consumerGroup, shardId, cursor);

```
}
```

Authorize a RAM user to access consumer groups

Before you use a RAM user to access consumer groups, you must grant the required permissions to the RAM user. For more information, see Grant permissions to a RAM role.

The following table describes the actions that a RAM user can perform.

Action	Description	Resource
log:GetCursorOrData (GetCursor and PullLogs)	Obtains a cursor based on the point in time when log data is generated.	acs:log: <i>\${regionName}:\${projectOwnerAliUid}</i> :proje ct/ <i>\${projectName}</i> /logstore/ <i>\${logstoreName}</i>
log:CreateConsumerGroup	Creates a consumer group for a specified Logstore.	acs:log:\${regionName}:\${projectOwnerAliUid}:proje ct/\${projectName}/logstore/\${logstoreName}/consu mergroup/*
log:ListConsumerGroup	Queries all consumer groups in a specified Logstore.	acs:log:\${regionName}:\${projectOwnerAliUid}:proje ct/\${projectName}/logstore/\${logstoreName}/consu mergroup/*
log:UpdateCheckPoint	Updates the consumption checkpoint in a shard of a specified consumer group.	acs:log:\${regionName}:\${projectOwnerAliUid}:proje ct/\${projectName}/logstore/\${logstoreName}/consu mergroup!\${consumerGroupName}
log:ConsumerGroupHeartBeat	Sends a heartbeat packet to Log Service for a specified consumer.	acs:log:\${regionName}:\${projectOwnerAliUid}:proje ct/\${projectName}(logstore \${logstoreName}/consu mergroup/\${consumerGroupName}
log:UpdateConsumerGroup	Modifies the attributes of a specified consumer group.	acs:log:\${regionName}:\${projectOwnerAliUid}:proje ct/\${projectName}(logstore \${logstoreName}/consu mergroup/\${consumerGroupName}
log:ConsumerGroupUpdateCheckPoint	Retrieves the consumption checkpoints in one or all shards of a specified consumer group.	acs:log:\${regionName}:\${projectOwnerAliUid}:proje ct/\${projectName}/logstore \${logstoreName}/consu mergroup!\${consumerGroupName}

For example, the project-test project resides in the China (Hangzhou) region. The ID of the Apsara Stack tenant account to which the project belongs is 174649****602745. The name of the Logstore from which you want to consume log data is logstore-test, and the consumer group name is consumergroup-test. To allow a RAM user to access the consumer group, you must grant the following permissions to the RAM user:

{	
	Jersion": "1",
":	Statement": [
	{
	"Effect": "Allow",
	"Action": [
	"log:GetCursorOrData"
	1,
	"Resource": "acs:log:cn-hangzhou:174649****602745:project/project-test/logstore/logstore-test"
	},
	{
	"Effect": "Allow",
	"Action": [
	"log:CreateConsumerGroup",
	"log:ListConsumerGroup"
],
	"Resource": "acs:log:cn-hangzhou:174649****602745:project/project-test/logstore/logstore-test/consumergroup/*"
	},
	{
	"Effect": "Allow",
	"Action": [
	"log:ConsumerGroupUpdateCheckPoint",
	"log:ConsumerGroupHeartBeat",
	"log:UpdateConsumerGroup",
	"log:GetConsumerGroupCheckPoint"
],
	"Resource": "acs:log:cn-hangzhou:174649****602745:project/project-test/logstore/logstore-test/consumergroup/consumergroup-test"
	}
]	
}	

4.7.3.2. View the status of a consumer group

This topic describes how to view the status of a consumer group.

View the consumption progress in the Log Service console

- 1. Log on to the Log Service console.
- 2. In the Projects section, click the project that you want to manage.
- 3. In the Logstores list, find the Logstore that you want to manage and choose > Data Consumption.
- 4. Click the consumer group whose data consumption progress you want to view. The data consumption progress of each shard in the Logstore is displayed.

Call the API or use an SDK to view the data consumption progress

The following code shows how to call the API to view the data consumption progress. In this example, Log Service SDK for Java is used.

package test; import java.util.ArrayList; import com.aliyun.openservices.log.Client; import com.aliyun.openservices.log.common.Consts.CursorMode; import com.aliyun.openservices.log.common.ConsumerGroup; import com.aliyun.openservices.log.common.ConsumerGroupShardCheckPoint; import com.aliyun.openservices.log.exception.LogException; public class ConsumerGroupTest { static String endpoint = ""; static String project = ""; static String logstore = ""; static String accesskeyId = ""; static String accesskey = ""; public static void main(String[] args) throws LogException { Client client = new Client (endpoint, accesskeyId, accesskey); // Obtain all consumer groups that are created for the Logstore. If no consumer group exists, an empty string is returned. List<ConsumerGroup> consumerGroups = client.ListConsumerGroup(project, logstore).GetConsumerGroups(); for(ConsumerGroup c: consumerGroups) { // Print the properties of consumer groups. The properties of a consumer group include the name, heartbeat timeout, and whether the consumer group consumes data in order. System.out.println("Name: " + c.getConsumerGroupName()); System.out.println("Heartbeat timeout: " + c.getTimeout()); System.out.println("Consumption in order: " + c.isInOrder()); for(ConsumerGroupShardCheckPoint cp: client.GetCheckPoint(project, logstore, c.getConsumerGroupName()).GetCheckPoints()){ System.out.println("shard: " + cp.getShard()); // The consumption time. The time is a long integer and is accurate to milliseconds. System.out.println("The last time when data was consumed: " + cp.getUpdateTime()); System.out.println("Consumer name: " + cp.getConsumer()); String consumerPrg = ""; if(cp.getCheckPoint().isEmpty()) consumerPrg = "Consumption not started"; else{ // The UNIX timestamp. Unit: seconds. Format the output value of the timestamp. try{ int prg = client.GetPrevCursorTime(project, logstore, cp.getShard(), cp.getCheckPoint()).GetCursorTime(); consumerPrg = "" + prg; catch(LogException e) { if(e.GetErrorCode() == "InvalidCursor") consumerPrg = "Invalid. The previous point in time when data was consumed is out of the retention period of the dat a in the Logstore"; else{ //internal server error throw e; } } 1 System.out.println("Consumption progress: " + consumerPrg); String endCursor = client.GetCursor(project, logstore, cp.getShard(), CursorMode.END).GetCursor(); int endPrg = 0; try{ endPrg = client.GetPrevCursorTime(project, logstore, cp.getShard(), endCursor).GetCursorTime(); catch(LogException e) { //do nothing // The UNIX timestamp. Unit: seconds. Format the output value of the timestamp. System.out.println("The point in time when the last data entry was received: " + endPrq); } } }

4.7.4. Use Storm to consume log data

Log Service LogHub provides efficient and reliable log collection and consumption channels. You can use services such as Logtail or SDKs to collect log data in real time. After log data is collected and sent to Log Service, you can use stream computing systems such as Spark Streaming and Apache Storm to consume the log data.

To reduce the costs of data consumption, Log Service provides LogHub Storm spouts to read data from Log Service in real time.

Architecture and implementation

- In the following figure, LogHub Storm spouts are enclosed in red dashed-line boxes. Each Storm topology has a group of spouts that work together to read data from a Logstore. Spouts in different topologies are independent of each other.
- Each topology is identified by the unique name of a LogHub consumer group. Spouts in the same topology use a consumer group to perform load balancing and automatic failover. For more information, see Use consumer groups to consume log data.
- Spouts in a topology read data from a Logstore in real time, and then send the data to bolts in the topology. The spouts save consumption checkpoints to the LogHub server on a regular basis.

Figure 1. Architecture and implementation



Limits

- You can create a maximum of 10 consumer groups to consume log data from a Logstore. If you no longer need a consumer group, you can call the DeleteConsumerGroup operation of the SDK for Java to delete the consumer group.
- We recommend that you configure the same number of spouts for a Logstore as the number of shards in the Logstore. This is because a single spout may not be able to process a large amount of data from multiple shards.
- If the data volume in a shard exceeds the processing capacity of a single spout, you can split the shard to reduce the data volume in each shard.
- LogHub Storm spouts and bolts must use the ack method to check whether log data is sent from spouts to bolts and whether the data is processed by the bolts.

Examples

The following code provides an example to show how to construct a Storm topology:

public static void main (String[] args) String mode = "Local"; // Use the local test mode. String consumser_group_name = ""; // The name of the consumer group of a topology. The name must be unique. The name cannot be an empty string. The name must be 3 to 63 characters in length and can contain lowercase letters, digits, hyphens (-), and underscores (_). The name must start and end with a lowercase letter or a digit. String project = ""; // The Log Service project. String logstore = ""; // The Log Service Logstore. String endpoint = ""; // The endpoint of Log Service. String access_id = ""; // The AccessKey ID. String access_key = ""; // Configure a LogHub Storm spout. LogHubSpoutConfig config = new LogHubSpoutConfig(consumser group name, endpoint, project, logstore, access_id, access_key, LogHubCursorPosition.END_CURSOR); TopologyBuilder builder = new TopologyBuilder(); // Create a LogHub Storm spout. LogHubSpout spout = new LogHubSpout(config); // In actual scenarios, we recommend that you create the same number of spouts for a Logstore as the number of shards in the Logstore. builder.setSpout("spout", spout, 1); builder.setBolt("exclaim", new SampleBolt()).shuffleGrouping("spout"); Config conf = new Config(); conf.setDebug(false); conf.setMaxSpoutPending(1); // If you use Kryo to serialize and deserialize data, configure the serialization method of LogGroupData by using the LogGroupDataSer ializSerializer class. Config.registerSerialization(conf, LogGroupData.class, LogGroupDataSerializSerializer.class); if (mode.equals("Local")) { logger.info("Local mode..."); LocalCluster cluster = new LocalCluster(); cluster.submitTopology("test-jstorm-spout", conf, builder.createTopology()); try { Thread.sleep(6000 * 1000); //waiting for several minutes } catch (InterruptedException e) { // TODO Auto-generated catch block e.printStackTrace(); cluster.killTopology("test-jstorm-spout"); cluster.shutdown(); } else if (mode.equals("Remote")) { logger.info("Remote mode..."); conf.setNumWorkers(2); try { StormSubmitter.submitTopology("stt-jstorm-spout-4", conf, builder.createTopology()); } catch (AlreadyAliveException e) // TODO Auto-generated catch block e.printStackTrace(); } catch (InvalidTopologyException e) { // TODO Auto-generated catch block e.printStackTrace(); } else { logger.error("invalid mode: " + mode); } }

• The following code provides an example to show how to consume log data, and then display the content of each log entry by using bolts:

```
public class SampleBolt extends BaseRichBolt {
   private static final long serialVersionUID = 4752656887774402264L;
    private static final Logger logger = Logger.getLogger(BaseBasicBolt.class);
    private OutputCollector mCollector;
    @Override
    public void prepare(@SuppressWarnings("rawtypes") Map stormConf, TopologyContext context,
           OutputCollector collector) {
        mCollector = collector;
    @Override
    public void execute(Tuple tuple) {
        String shardId = (String) tuple
                .getValueByField(LogHubSpout.FIELD SHARD ID);
        @SuppressWarnings("unchecked")
        List<LogGroupData> logGroupDatas = (ArrayList<LogGroupData>) tuple.getValueByField(LogHubSpout.FIELD_LOGGROUPS);
        for (LogGroupData groupData : logGroupDatas) {
            // Each log group consists of one or more log entries.
            LogGroup logGroup = groupData.GetLogGroup();
            for (Log log : logGroup.getLogsList()) {
                StringBuilder sb = new StringBuilder();
                \ensuremath{\prime\prime}\xspace Lach log entry contains a time field and other fields that are formatted in key-value pairs.
                int log time = log.getTime();
                sb.append("LogTime:").append(log time);
                for (Content content : log.getContentsList()) {
                    sb.append("\t").append(content.getKey()).append(":")
                            .append(content.getValue());
                logger.info(sb.toString());
           }
        // LogHub spouts and bolts must use the ack method to check whether log data is sent from spouts to bolts and whether the data is
processed by the bolts.
       mCollector.ack(tuple);
    @Override
    public void declareOutputFields(OutputFieldsDeclarer declarer) {
       //do nothing
    }
}
```

Maven

The following code provides an example to show how to add Maven dependencies for Storm 1.0 or earlier, such as Storm 0.9.6:

```
<dependency>
<groupId>com.aliyun.openservices</groupId>
<artifactId>loghub-storm-spout</artifactId>
<version>0.6.6</version>
</dependency>
```

The following code provides an example to show how to add Maven dependencies for Storm 1.0 or later:

<dependency>

```
<groupId>com.aliyun.openservices</groupId>
<artifactId>loghub-storm-1.0-spout</artifactId>
<version>0.1.3</version>
</dependency>
```

4.7.5. Use Flume to consume log data

This topic describes how to use Flume to consume log data. You can use the aliyun-log-flume plug-in to connect Log Service to Flume and write log data to Log Service or consume log data from Log Service.

Background information

The aliyun-log-flume plug-in connects Log Service to Flume. When Log Service is connected to Flume, you can use Flume to connect Log Service to other systems such as Hadoop distributed file system (HDFS) and Kafka. The aliyun-log-flume plug-in provides sinks and sources to connect Log Service to Flume.

- Sink: reads data from other data sources and writes the data to Log Service.
- Source: consumes log data from Log Service and writes the log data to other systems.
- For more information, visit GitHub.

Procedure

- 1. Download and install Flume. For more information, see Flume.
- 2. Download the aliyun-log-flume plug-in and save the plug-in in the cd/***/flume/lib directory. To download the plug-in, click aliyun-log-flume-1.3.jar.
- 3. In the cd/***/flume/conf directory, create a configuration file named flumejob.conf.
 - For information about how to configure a sink, see Sink.
 - For information about how to configure a source, see Source.

4. Start Flume.

Sink

You can configure a sink for Flume to write data from other data sources to Log Service. Data can be parsed into the following two formats:

• SIMPLE: A Flume event is written to Log Service as a field.

• DELIMITED: A Flume event is parsed into fields based on the configured column names and written to Log Service. The following table describes the parameters of a sink.

Parameter	Required	Description
type	Yes	Default value: com.aliyun.Loghub.flume.sink.LoghubSink .
endpoint	Yes	The endpoint of the region where the Log Service project resides.
project	Yes	The name of the project.
logstore	Yes	The name of the Logstore.
accessKeyId	Yes	The AccessKey ID that is used to access Log Service. For more information, see Obtain an AccessKey pair.
accessKey	Yes	The AccessKey secret that is used to access Log Service. For more information, see Obtain an AccessKey pair.
batchSize	No	The number of data entries that are written to Log Service at a time. Default value: 1000.
maxBufferSize	No	The maximum number of data entries in the buffer. Default value: 1000.
serializer	No	 The serialization format of the Flume event. Default value: SIMPLE. Valid values: DELIMITED: delimiter mode. SIMPLE: single-line mode. Custom serializer: custom serialization mode. In this mode, you must specify the full names of columns.
columns	No	The names of colums. If you set the serializer parameter to DELIMITED , you must specify this parameter. Separate multiple columns with commas (,). The columns are sorted in the same order that the columns are sorted in the data entries.
separatorChar	No	The delimiter. If you set the serializer parameter to DELIMITED , you must specify a single character for this parameter. The default value is a comma (,).
quoteChar	No	The quote character. If you set the serializer parameter to DELIMITED , you must specify this parameter. The default value is double quotation marks (").
escapeChar	No	The escape character. If you set the serializer parameter to DELIMITED , you must specify this parameter. The default value is double quotation marks (").
useRecordTime	No	Specifies whether to use the value of the timestamp field in the data entries as the log time when data is written to Log Service. Default value: false. This value indicates that the current time is used as the log time.

For more information about how to configure a sink, visit GitHub.

Source

You can configure a source for Flume to ship data from Log Service to other data sources. Data can be parsed into the following two formats: • DELIMITED: Log data is written to Flume in delimiter mode.

JSON: Log data is written to Flume in the JSON format.

The following table describes the parameters of a source.

Parameter	Required	Description
type	Yes	Default value: com.aliyun.loghub.flume.source.LoghubSource .
endpoint	Yes	The endpoint of the region where the Log Service project resides.
project	Yes	The name of the project.
logstore	Yes	The name of the Logstore.
accessKeyId	Yes	The AccessKey ID that is used to access Log Service. For more information, see Obtain an AccessKey pair.
accessKey	Yes	The AccessKey secret that is used to access Log Service. For more information, see Obtain an AccessKey pair.
heartbeatIntervalMs	No	The heartbeat interval between the client and Log Service. Default value: 30000. Unit: milliseconds.
fetchIntervalMs	No	The interval at which data is pulled from Log Service. Default value: 100. Unit: milliseconds.
fetchInOrder	No	Specifies whether to consume log data in the order that the log data is written to Log Service. Default value: false.
batchSize	No	The number of log entries that are read at a time. Default value: 100.
consumerGroup	No	The name of the consumer group that reads log data.
initialPosition	No	The start point from which data is read. Valid values: begin , end , and timestamp . Default value: begin .

timestamp	No	The UNIX timestamp. If you set the initialPosition parameter to timestamp , you must specify this parameter.
deserializer	Yes	 The deserialization format of the Flume event. Default value: DELIMITED. Valid values: DELIMITED: delimiter mode. JSON: JSON format. Custom deserializer: custom deserialization mode. In this mode, you must specify the full names of the columns.
columns	No	The names of colums. If you set the deserializer parameter to DELIMITED , you must specify this parameter. Separate multiple columns with commas (,). The columns are sorted in the same order that the columns are sorted in the log entries.
separatorChar	No	The delimiter. If you set the deserializer parameter to DELIMITED , you must specify a single character for this parameter. The default value is a comma (,).
quoteChar	No	The quote character. If you set the deserializer parameter to DELIMITED , you must specify this parameter. The default value is double quotation marks (").
escapeChar	No	The escape character. If you set the deserializer parameter to DELIMITED , you must specify this parameter. The default value is double quotation marks (").
appendTimestamp	No	Specifies whether to append the timestamp as a field to the end of each log entry. If you set the deserializer parameter to DELIMITED , you must specify this parameter. Default value: false.
sourceAsField	No	Specifies whether to add the log source as a field namedsource If you set the deserializer parameter to JSON , you must specify this parameter. Default value: false.
tagAsField	No	Specifies whether to add the log tag as a field named <u>tag</u> :{name of the tag}. If you set the deserializer parameter to JSON , you must specify this parameter. Default value: false.
timeAsField	No	Specifies whether to add the log time as a field namedtime if you set the deserializer parameter to JSON , you must specify this parameter. Default value: false.
useRecordTime	No	Specifies whether to use the value of the timestamp field in the log entries as the log time when log data is read from Log Service. Default value: false. This value indicates that the current time is used as the log time.

For more information about how to configure a source, visit GitHub.

4.7.6. Use open source Flink to consume log data

Log Service provides the flink-log-connector agent to connect with Flink. This topic describes how to connect Log Service with Flink to consume log data.

Prerequisites

- An AccessKey pair is obtained. For more information, see Obtain an AccessKey pair.
- A project and a Logstore are created. For more information, see Create a project and Create a Logstore.
- A RAM user is authorized to access the Logstore from which you want to consume data. For more information, see Grant a RAM user the permissions to consume data from a specified Logstore.

Background information

The flink-log-connector agent consists of the flink-log-consumer and flink-log-producer agents. The two agents have the following differences:

- The flink-log-consumer agent reads data from Log Service. This agent supports the exactly-once semantics and load balancing among shards.
 The flink-log-producer agent writes data to Log Service.
- Before you can use the flink-log-producer agent to write data to Log Service, you must add Maven dependencies. The following example shows sample Maven dependencies:

```
<dependency>
<groupId>com.aliyun.openservices//groupId>
<artifactId>flink-log-connector</artifactId>
</dependency>
<dependency>
<groupId>com.google.protobuf</groupId>
<artifactId>protobuf-java</artifactId>
</version>2.5.0</version>
</dependency>
```

For more information, visit GitHub.

Flink Log Consumer

The flink-log-consumer agent can consume log data from a Logstore based on the exactly-once semantics. The flink-log-consumer agent detects the change in the number of shards in a Logstore.

Each Flink subtask consumes data from some shards in a Logstore. If the shards in a Logstore are split or merged, the shards from which the subtask consumes data also change.

If you use the flink-log-consumer agent to consume data from Log Service, you can call the following API operations:

GetCursorOrData

You can call this operation to pull log data from a shard. If you frequently call this operation, the amount of data that is transferred may exceed the capacity of shards. You can use the **ConfigConstants.LOG_FETCH_DATA_INTERVAL_MILLIS** parameter to specify the interval of API calls. You can use the **ConfigConstants.LOG_MAX_NUMBER_PER_FETCH** parameter to specify the number of log entries pulled by each call. Example:

configProps.put(ConfigConstants.LOG_FETCH_DATA_INTERVAL_MILLIS, "100"); configProps.put(ConfigConstants.LOG MAX NUMBER PER FETCH, "100");

- ListShards
- You can call this operation to view all shards in a Logstore and the status of each shard. If the shards are frequently split and merged, you can adjust the call interval to detect the changes in the number of shards at the earliest opportunity. Example

// Call the ListShards operation every 30,000 milliseconds.

configProps.put(ConfigConstants.LOG_SHARDS_DISCOVERY_INTERVAL_MILLIS, "30000");

CreateConsumerGroup

You can call this operation to create a consumer group that is used to synchronize checkpoints.

ConsumerGroupUpdateCheckPoint

You can call this operation to synchronize snapshots of Flink to a consumer group.

- 1. Set startup parameters
 - The following example shows how to consume log data. The java util Properties class is used as a configuration tool. All constants must be configured in the ConfigConstants class.
 - Properties configProps = new Properties();
 - // Specify the endpoint of Log Service.
 - configProps.put(ConfigConstants.LOG_ENDPOINT, "cn-hangzhou.log.aliyuncs.com");
 - // Specify the AccessKey ID and AccessKey secret.
 - configProps.put(ConfigConstants.LOG_ACCESSKEYID, ""); configProps.put(ConfigConstants.LOG_ACCESSKEY, "");
 - // Specify the project

configProps.put(ConfigConstants.LOG_PROJECT, "ali-cn-hangzhou-sls-admin");

- // Specify the Logstore
- configProps.put(ConfigConstants.LOG LOGSTORE, "sls consumergroup log");
- // Specify the start position of log consumption
- configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, Consts.LOG_END_CURSOR);
- // Specify the data deserialization method.
- RawLogGroupListDeserializer deserializer = new RawLogGroupListDeserializer();
- final StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();
- DataStream<RawLogGroupList> logTestStream = env.addSource(
 - new FlinkLogConsumer<RawLogGroupList>(deserializer, configProps));

② Note The number of Flink subtasks is independent of the number of shards in a Logstore. If the number of shards is greater than the number of subtasks, each subtask consumes one or more shards. If the number of shards is less than the number of subtasks, some subtasks remains idle until new shards are generated. The data of each shard is consumed by only one subtask.

Specify the start position of log consumption. 2.

- If you use the flink-log-consumer agent to consume data from a Logstore, you can use the **ConfigConstants.LOG_CONSUMER_BEGIN_POSITION** parameter to specify the start position of log consumption. You can start to consume data from the earliest entry, the latest entry, or from a specific point in time. The flink-log-consumer agent also allows you to resume consumption from a specific consumer group. You can set the parameter to one of the following values:
- $\circ~$ Consts.LOG_BEGIN_CURSOR: starts to consume data from the earliest entry.
- · Consts.LOG END CURSOR: starts to consume data from the latest entry
- Consts.LOG_FROM_CHECKPOINT: starts to consume data from a checkpoint that is stored in a specified consumer group. You can use the ConfigConstants.LOG_CONSUMERGROUP parameter to specify the consumer group.
- UnixTimestamp: a string of the integer data type. The timestamp follows the UNIX time format. It is the number of seconds that have elapsed since 00:00:00 Thursday, January 1, 1970. The value indicates that the data in a shard is consumed from this point in time.

Example:

configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, Consts.LOG_BEGIN_CURSOR); configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, Consts.LOG_END_CURSOR); configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, "1512439000"); configProps.put(ConfigConstants.LOG_CONSUMER_BEGIN_POSITION, Consts.LOG_FROM_CHECKPOINT);

() Note If you configure to resume consumption from a state backend of Flink when you start a Flink job, the flink-log-connector agent uses checkpoints that are stored in the state backend.

3. Optional: Configure consumption progress monitoring.

The flink-log-connector agent allows you to monitor the consumption progress. You can use the monitoring feature to obtain the consumption position of each shard in real time. The consumption position is indicated by a timestamp. For more information, see View the status of a consumer

Example:

configProps.put(ConfigConstants.LOG_CONSUMERGROUP, "your consumer group name");

② Note This setting is optional. If you configure consumption progress monitoring and no consumer group exists, the flink-log-connector agent creates a consumer group. If a consumer group is available, the agent synchronizes snapshots to the consumer group. You can view the consumption progress of the agent in the Log Service console.

4. Configure consumption resumption and the exactly-once semantics.

If the checkpointing feature of Flink is enabled, the flink-log-consumer agent periodically saves the consumption progress of each shard. If a subtask fails, Flink restores the subtask and starts to consume data from the latest checkpoint. When you specify the checkpoint period, you can specify the maximum amount of data that can be re-consumed if a subtask fails. You can use the

following code to specify the checkpoint period:

final StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();

// Configure the exactly-once semantics

env.getCheckpointConfig().setCheckpointingMode(CheckpointingMode.EXACTLY_ONCE);

// Save checkpoints every 5 s

env.enableCheckpointing(5000);

For more information, see Checkpoints.

Flink Log Producer

The flink-log-producer agent writes data to Log Service.

(2) Note The flink-log-producer agent supports only the Flink at-least-once semantics. If a subtask fails, duplicate data may be written to Log Service. However, no data is lost.

- If you use the flink-log-producer agent to write data to Log Service, you can call the following API operations:
- PostLogStoreLogs
- ListShards
- 1. Initialize the flink-log-producer agent.
- i. Set the initialization parameters of the flink-log-producer agent. The flink-log-producer agent is initialized the same way as the flink-log-consumer agent. The following example shows how to configure the initialization parameters of the flink-log-producer agent. In most cases, you can use the default values of the parameters. Example: // The number of I/O threads that are used to send data. Default value: 8. ConfigConstants.LOg_SENDER_IO_THREAD_COUNT
- // The time that is required to send cached logs. Default value: 3000. ConfigConstants.LOG_PACKAGE_TIMEOUT_MILLIS // The number of logs in the cached packet. Default value: 4096. ConfigConstants.LOG_LOGS_COUNT_PER_PACKAGE // The size of the cached packet. Default value: 3. Unit: MB. ConfigConstants.LOG_LOGS_BYTES_PER_PACKAGE // The total size of memory that can be used by the job. Default value: 100. Unit: MB. ConfigConstants.LOG_MEM_POOL_BYTES ii. Reload LogSerializationSchema and define the method that is used to serialize data into raw log groups. A raw log group is a collection of log entries.
 - If you want to write data to a specific shard, you can use the LogPartitioner parameter to generate hash keys for log data. LogPartitioner is an optional parameter. If you do not specify this parameter, data is written to a random shard.

Example:

2. Write simulated data to Log Service, as shown in the following example:
// Serialize data into the format of raw log groups. class SimpleLogSerializer implements LogSerializationSchema<String> { public RawLogGroup serialize(String element) { RawLogGroup rlg = new RawLogGroup(); RawLog rl = new RawLog(); rl.setTime((int)(System.currentTimeMillis() / 1000)); rl.addContent("message", element); rlg.addLog(rl); return rlg; public class ProducerSample { public static String sEndpoint = "cn-hangzhou.log.aliyuncs.com"; public static String sAccessKeyId = ""; public static String sAccessKey = ""; public static String sProject = "ali-cn-hangzhou-sls-admin"; public static String sLogstore = "test-flink-producer"; private static final Logger LOG = LoggerFactory.getLogger(ConsumerSample.class); public static void main(String[] args) throws Exception { final ParameterTool params = ParameterTool.fromArgs(args); final StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment(); env.getConfig().setGlobalJobParameters(params); env.setParallelism(3); DataStream<String> simpleStringStream = env.addSource(new EventsGenerator()); Properties configProps = new Properties(); // Specify the endpoint of Log Service. configProps.put(ConfigConstants.LOG_ENDPOINT, sEndpoint); // Specify the AccessKey ID and AccessKey secret. configProps.put(ConfigConstants.LOG ACCESSKEYID, sAccessKeyId); configProps.put(ConfigConstants.LOG_ACCESSKEY, sAccessKey); // Specify the project to which logs are written. configProps.put(ConfigConstants.LOG_PROJECT, sProject); // Specify the Logstore to which logs are written. configProps.put(ConfigConstants.LOG LOGSTORE, sLogstore); FlinkLogProducer<String> logProducer = new FlinkLogProducer<String>(new SimpleLogSerializer(), configProps); simpleStringStream.addSink(logProducer); env.execute("flink log producer"); // Simulate log generation. public static class EventsGenerator implements SourceFunction<String> { private boolean running = true; @Override public void run(SourceContext<String> ctx) throws Exception { long seg = 0; while (running) Thread.sleep(10); ctx.collect((seq++) + "-" + RandomStringUtils.randomAlphabetic(12)); } @Override public void cancel() { running = false; }

4.7.7. Use Logstash to consume log data

Log Service provides Logstash that you can use to consume log data. You can configure the Logstash input plug-in to read log data from Log Service, and then write the data to other systems, such as Kafka and Hadoop Distributed File System (HDFS).

Features

- Distributed collaborative consumption: You can configure multiple servers to consume log data from a Logstore at the same time.
- High performance: If you use a Java consumer group, the consumption speed of a single-core CPU can reach 20 MB/s.
- High reliability: Log Service saves consumption checkpoints. This mechanism resumes log consumption from the last checkpoint after a consumption exception is resolved.
- Automatic load balancing: Shards are automatically allocated based on the number of consumers in a consumer group. If you add a consumer to a
 consumer group or remove a consumer from the consumer group, shards are automatically reallocated.

Procedure

- 1. Install Logstash.
 - i. Download the Logstash installation package.
 - ii. Decompress the package that you downloaded to the specified directory.
- 2. Install the Logstash input plug-in.
 - i. Download the input plug-in logstash-input-sls.
 - ii. Install the Logstash input plug-in.

logstash-plugin install logstash-input-sls

② Note For information about the causes of installation failures and solutions, see Plug-in installation and configuration.

3. Start Logstash.

logstash -f logstash.conf

The following table describes the parameters of the Logstash input plug-in.

Parameter	Туре	Required	Description
endpoint	String	Yes	The endpoint of the region where the Log Service project resides.
access_id	String	Yes	The AccessKey ID of an Apsara Stack tenant account or RAM user that is used to access the project. The Apsara Stack tenant account or RAM user must have the permissions to consume log data by using consumer groups. For more information, see Permission to consume data of a specified Logstore.
access_key	String	Yes	The AccessKey secret of an Apsara Stack tenant account or RAM user that is used to access the project. The Apsara Stack tenant account or RAM user must have the permissions to consume log data by using consumer groups. For more information, see Permission to consume data of a specified Logstore
project	String	Yes	The name of the Log Service project.
logstore	String	Yes	The name of the Log Service Logstore.
consumer_group	String	Yes	The name of the consumer group.
consumer_name	String	Yes	The name of the consumer. The name of each consumer in a consumer group must be unique.
position	String	Yes	 The position where data consumption starts. Valid values: begin: Data is consumed from the first log entry that is written to the Logstore. end: Data is consumed from the current point in time. yyyy-MM-dd HH:mm:ss: Data is consumed from the specified point in time.
checkpoint_second	Number	No	The interval at which checkpoints are recorded. We recommend that you set the interval to a value between 10 and 60. Minimum value: 10. Default value: 30. Unit: seconds.
include_meta	Boolean	No	 Specifies whether input log data contains metadata, such as the log source, time, tags, and topic fields. Default value: true. true: The log data contains metadata. false: The log data does not contain metadata.
consumer_name_with _ip	Boolean	No	 Specifies whether to include an IP address in a consumer name. Default value: true. You must set this parameter to true if you want to apply distributed collaborative consumption. true: The name of the consumer contains an IP address. false: The name of the consumer does not contain an IP address.

Example

The following script shows how to configure Logstash to consume log data from a Logstore, and then print the data in stdout logs:

```
input {
  logservice{
  endpoint => "your project endpoint"
  access_id => "your access id"
  access_key => "your access key"
  project => "your project name"
  logstore => "your logstore name"
  consumer_group => "consumer group name"
  consumer_name => "consumer name"
  position => "end"
  checkpoint_second => 30
  include_meta => true
  consumer_name_with_ip => true
  }
 }
output {
  stdout {}
 }
```

4.7.8. Use Spark Streaming to consume log data

After Log Service collects log data, you can use Spark Streaming to consume the data.

The Spark SDK provided by Alibaba Cloud allows you to consume log data from Log Service in Receiver or Direct mode. You must add the following Maven dependency:

```
<dependency>
<groupId>com.aliyun.emr</groupId>
<artifactId>emr-logservice_2.11</artifactId>
<version>1.7.2</version>
</dependency>
```

Consume log data in Receiver mode

In Receiver mode, a consumer group consumes data from Log Service and temporarily stores the data in a Spark executor. After a Spark Streaming job is started, the consumer group reads and processes data from the Spark executor. Each log entry is returned as a JSON string. The consumer group

periodically saves checkpoints to Log Service. You do not need to update checkpoints. For more information, see Use consumer groups to consume log data.

Parameters

Parameter	Туре	Description
project	String	The name of the Log Service project.
logstore	String	The name of the Log Service Logstore.
consumerGroup	String	The name of the consumer group.
endpoint	String	The endpoint of the region where the Log Service project resides.
accessKeyId	String	The AccessKey ID that is used to access Log Service.
accessKeySecret	String	The AccessKey secret that is used to access Log Service.

Example

⑦ Note In Receiver mode, data loss may occur if the default configurations are used. To avoid data loss, you can enable the Write-Ahead Logs feature. This feature is available in Spark 1.2 or later. For more information, see Spark.
<pre>import org.apache.spark.storage.StorageLevel import org.apache.spark.streaming.aliyun.logservice.LoghubUtils import org.apache.spark.streaming.{Milliseconds, StreamingContext} import org.apache.spark.SparkConf</pre>
<pre>object TestLoghub { def main(args: Array[String]): Unit = { if (args.length < 7) {</pre>
System.err.println("""Usage: TestLoghub <project> <logstore> <loghub group="" name=""> <endpoint> caccess key id> <access key="" secret=""> <batch interval="" seconds=""> """.stripMarqin)</batch></access></endpoint></loghub></logstore></project>
System.exit(1) }
<pre>val project = args(0) val postcre = args(1) val consumerGroup = args(2) val endpoint = args(3)</pre>
<pre>val accessKeyId = args(4) val accessKeySecret = args(5) val batchInterval = Milliseconds(args(6).toInt * 1000)</pre>
<pre>def functionToCreateContext(): StreamingContext = { val conf = new SparkConf().setAppName("Test Loghub") val ssc = new StreamingContext(conf, batchInterval)</pre>
val loghubStream = LoghubUtils.createStream(ssc, project, logstore,
consumerGroup, endpoint, accessKeyId,
<pre>loghubStream.checkpoint(batchInterval * 2).foreachRDD(rdd =></pre>
<pre>rdd.map(bytes => new String(bytes)).top(10).foreach(println)) ssc.checkpoint("hdfs:///tmp/spark/streaming") // set checkpoint directory ssc</pre>
} val ssc = StreamingContext.getOrCreate("hdfs:///tmp/spark/streaming", functionToCreateContext _)
<pre>ssc.start() ssc.awaitTermination() }</pre>
}

Consume log data in Direct mode

In Direct mode, you can consume log data from Log Service without the need of consumer groups. You can call API operations to request data from Log Service. Consuming log data in Direct mode has the following benefits:

- Simplified concurrency. The number of Spark partitions is the same as the number of shards in a Logstore. You can split shards to improve the concurrency of tasks.
- Increased efficiency. You no longer need to enable the Write-Ahead Logs feature to prevent data loss.
- Exactly-once semantics. Data is directly read from Log Service. Checkpoints are submitted after a task is successful.

In some cases, data may be repeatedly consumed if a task ends due to an unexpected exit of Spark.

If you want to consume data in Direct mode, you must configure the ZooKeeper service to store intermediate data. In addition, you must specify a checkpoint directory in the ZooKeeper service. Intermediate data is stored in the checkpoint directory. To re-consume data after you restart a task, you must delete the checkpoint directory from ZooKeeper and change the name of the consumer group.

• Parameters

Parameter	Туре	Description
project	String	The name of the Log Service project.
logstore	String	The name of the Log Service Logstore.
consumerGroup	String	The name of the consumer group. This name is used only to save consumption checkpoints.
endpoint	String	The endpoint of the region where the Log Service project resides.
accessKeyId	String	The AccessKey ID that is used to access Log Service.
accessKeySecret	String	The AccessKey secret that is used to access Log Service.
zkAddress	String	The connection URL of the ZooKeeper service.

Throttling configuration

Spark Streaming consumes data from each shard in a single batch. You must specify the number of log entries that are consumed in each batch.

In the underlying storage model of Log Service, a log group serves as the basic storage unit. Each log group corresponds to a write request. For example, a write request may contain multiple log entries. These log entries are stored and consumed as a log group. When you use web tracking to write logs, each write request contains only one log entry. In this case, the log group that corresponds to the request contains only one log entry. You can specify parameters to limit the amount of log data in a single batch. The following table describes the two parameters.

Parameter	Description	Default
spark.loghub.batchGet.step	The maximum number of log groups that are returned for a single consumption request.	100
spark.streaming.loghub.maxRatePerShard	The maximum number of log entries that are consumed from each shard in a single batch.	10000

You can set the **spark.streaming.loghub.maxRatePerShard** parameter to specify the maximum number of log entries that are consumed from each shard in each batch. The Spark SDK obtains the number of log groups from the **spark.loghub.batchGet.step** parameter before it consumes log data from Log Service, and accumulates the number of log entries in these log groups during the consumption. When the accumulated number reaches or exceeds the specified number in the **spark.streaming.loghub.maxRatePerShard** parameter, the Spark SDK stops consuming log data. The **spark.streaming.loghub.maxRatePerShard** parameter does not precisely control the number of consumed log entries in each batch. The number of consumed log entries in each batch varies based on the value of the **spark.loghub.batchGet.step** parameter and the number of log entries in each log group.

Example

User Guide-Log Service

 $import\ {\tt com.aliyun.openservices.loghub.client.config.LogHubCursorPosition}$ import org.apache.spark.SparkConf import org.apache.spark.streaming.{Milliseconds, StreamingContext} import org.apache.spark.streaming.aliyun.logservice.{CanCommitOffsets, LoghubUtils} object TestDirectLoghub { def main(args: Array[String]): Unit = { if (args.length < 7) { System.err.println("""Usage: TestDirectLoghub <project> <logstore> <loghub group name> <endpoint> <access key id> <access key secret> <batch interval seconds> <zookeeper host:port=localhost:2181> """.stripMargin) System.exit(1) val project = args(0) val logstore = args(1) val consumerGroup = args(2) val endpoint = args(3) val accessKeyId = args(4) val accessKeySecret = args(5) val batchInterval = Milliseconds(args(6).toInt * 1000) val zkAddress = if (args.length >= 8) args(7) else "localhost:2181" def functionToCreateContext(): StreamingContext = { val conf = new SparkConf().setAppName("Test Direct Loghub") val ssc = new StreamingContext(conf, batchInterval) val zkParas = Map("zookeeper.connect" -> zkAddress, "enable.auto.commit" -> "false") val loghubStream = LoghubUtils.createDirectStream(ssc, project, logStore, consumerGroup, accessKeyId, accessKeySecret, endpoint. zkParas, LogHubCursorPosition.END CURSOR) loghubStream.checkpoint(batchInterval).foreachRDD(rdd => { println(s"count by key: \${rdd.map(s => { s.sorted (s.length, s) }).countByKey().size}") loghubStream.asInstanceOf[CanCommitOffsets].commitAsync() }) ssc.checkpoint("hdfs:///tmp/spark/streaming") // set checkpoint directory ssc } val ssc = StreamingContext.getOrCreate("hdfs:///tmp/spark/streaming", functionToCreateContext _) ssc.start() ssc.awaitTermination() } }

For more information, visit GitHub.

4.7.9. Use Realtime Compute to consume log data

You can use Realtime Compute (Blink) to create a schema for data in Log Service and consume the data. This topic describes how to use Realtime Compute to create a schema for data in Log Service. This topic also describes the attribute fields and data type mapping that you can configure when you create a schema.

Create a schema for data in Log Service

Log Service stores streaming data. Realtime Compute can use the streaming data as input data. The following example is a sample log entry:

```
__source_: 203.0.113.10
__tag_:_receive_time_: 1562125591
__topic_: test-topic
a: 1234
b: 0
c: hello
```

The following example is a DDL statement that is used to create a schema for data in Log Service:

create table sls_stream(a int, b int, c varchar) with (type ='sls', endPoint ='your AccessKey ID', accessKey ='your AccessKey Secret', startTime = '2017-07-05 00:00:00', project ='ali-cloud-streamtest', logStore ='stream-test', consumerGroup ='consumerGroupTest1');

The following table describes the parameters in the WITH clause.

Parameter	Required	Description
endPoint	Yes	The endpoint of Log Service. For more information, see Obtain an endpoint in Log Service Developer Guide.
accessId	Yes	The AccessKey ID that is used to access Log Service.
accessKey	Yes	The AccessKey secret that is used to access Log Service.
project	Yes	The name of the Log Service project.
logStore	Yes	The name of the Log Service Logstore.
consumerGroup	No	The name of the consumer group.
startTime	No	The point in time when Realtime Compute starts to consume log data.
heartBeatIntervalMills	No	The heartbeat interval of the client that consumes log data. Default value: 10. Unit: seconds.
maxRetryTimes	No	The maximum number of retries to read data. Default value: 5.
batchGetSize	No	The number of log groups that you want to read at a time. Default value: 10. If the version of Blink is 1.4.2 or later, the default value is 100 and the maximum value is 1000.
columnErrorDebug	No	Specifies whether to enable debugging. If you set the value to true, debugging is enabled and log entries that fail to be parsed are displayed. Default value: false. This value indicates that debugging is not enabled.

Attribute fields

Realtime Compute can extract fields from log data. Realtime Compute can also extract three attribute fields and custom tag fields. The following table describes the three attribute fields.

Attribute field	Description
source	The source of the log entry.
topic	The topic of the log entry.
timestamp	The point in time when the log entry is generated.

To extract the three attribute fields, you must add HEADERs in the DDL statement. Example:

```
create table sls_stream(
   __timestamp__ bigint HEADER,
   __receive_time__ bigint HEADER
a int,
b int,
c varchar
) with (
  type ='sls',
  endPoint ='your endpoint',
  accessId ='your AccessKey ID',
  accessKey ='your AccessKey Secret',
  startTime = '2017-07-05 00:00:00',
  project ='ali-cloud-streamtest',
  logStore ='stream-test',
  consumerGroup ='consumerGroupTest1'
```

);

Data type mapping

The string data type in Log Service is mapped to the varchar data type in Realtime Compute. We recommend that you declare the mapping in a DDL statement. If you specify another data type to convert data in Log Service, Realtime Compute attempts to automatically convert the data. For example, you can specify bigint as the data type to convert the string 1000 and specify timestamp as the data type to convert the string 2018-01-12 12:00:00 .

Usage notes

- Blink 2.2.0 or earlier versions do not support shard scaling. If you split or merge shards when a job is reading data from a Logstore, the job fails and cannot continue. In this case, you must restart the job.
- You cannot delete or recreate a Logstore whose log data is being consumed, regardless of the Blink version.
- In Blink version 1.6.0 and earlier, the read performance may be affected if you specify a consumer group to consume log data from a Logstore that contains a large number of shards.
- You cannot define the map data type in Realtime Compute when you create a schema for data in Log Service.
- · Fields that do not exist are set to null.
- Fields can be converted in a random order. However, we recommend that you convert the fields in the same order as the fields in the schema.
- If no new data is written to a shard, the latency of a job increases. In this case, you must change the number of concurrent tasks in the job to the number of shards in which data is read and written.
- To extract fields from tags such as __tag_:_hostname__ and __tag_:_path__ , you can delete the __tag_: prefix and follow the method used to extract attribute fields.

⑦ Note You cannot extract this type of data during debugging. We recommend that you use the on-premises debugging method and the print method to display data in logs.

4.8. Data shipping

4.8.1. Ship logs to OSS

4.8.1.1. Overview

Log Service provides the data shipping feature. You can use this feature to ship logs to Object Storage Service (OSS) in real time by using the Log Service console. This topic describes the benefits and scenarios of the data shipping feature.

In the Log Service console, you can ship logs to other Apsara Stack services. Then, you can store or consume the log data by using other systems such as E-MapReduce. After you enable the log shipping feature, Log Service ships the collected logs to the specified cloud service at regular intervals.

Scenarios

The data shipping feature can be used to connect Log Service with data warehouses.

Benefits

The data shipping feature of Log Service has the following benefits:

• Ease of use

You only need to complete a few settings in the Log Service console before you can ship logs from Logstores to other Apsara Stack services such as OSS.

High efficiency

Log Service stores logs that are collected from multiple servers. This improves efficiency when you ship log data to Apsara Stack services such as OSS.

Effective management

You can ship logs from different projects or Logstores to different OSS buckets. This way, you can efficiently manage the logs by log type or log source.

Log shipping destinations

For information about how to ship logs to OSS, see Ship log data from Log Service to OSS.

4.8.1.2. Ship log data from Log Service to OSS

You can use Log Service to collect log data and ship the log data to Object Storage Service (OSS) for storage and analysis. This topic describes how to ship log data from Log Service to OSS.

Prerequisites

- Log data is collected. For more information, see Log collection methods.
- OSS is activated. A bucket is created in the region where the Log Service project resides. For more information, see the **Create buckets** section in the **Service User Guide -Ojbect Storage Service(OSS)**.
- A Resource Access Management (RAM) role is created for the level-1 organization. For more information, see Obtain the ARN of a RAM role.

Background information

Log Service can automatically ship log data from a Logstore to an OSS bucket.

- You can set a custom retention period for the log data in the OSS bucket. Permanent retention is supported.
- You can use data processing platforms such as E-MapReduce and Data Lake Analytics (DLA) or use custom programs to consume log data from the OSS bucket.

with a lowercase letter or digit and must be 2 to 128 characters in length

Procedure

- 1. Log on to the Log Service console.
- 2. In the Projects section, click the project from which you want to ship log data to OSS.
- On the Logstores tab, click the > icon on the left of the specific Logstore and choose Data Transformation > Export > Object Storage Service(OSS).
- 4. On the OSS Shipper page, click Enable.
- 5. In the OSS LogShipper pane, configure the shipping rules.
 - Parameter
 Description

 OSS Shipper Name
 The name of the shipping rule.

OSS Bucket	The name of the OSS bucket to which you want to ship log data. () Important You must specify the name of an existing OSS bucket. The specified OSS bucket must reside in the same region as the Log Service project.		
OSS Prefix	The directory to which log data is shipped in the OSS bucket.		
Partition format	The partition format of the bucket directory for the shipping task. The directory is automatically generated based on the time when the shipping task is created. The default format is %Y/%m/%d/%H/%M. The partition format cannot start with a forward slash (/). For information about partition format examples, see Partition format. For more information about parameters, see strptime API.		
(Resource Access Management) RAM Role	The Alibaba Cloud Resource Name (ARN) of the RAM role. The RAM role is the identity that the OSS bucket owner creates for access control. Example: acs:ram::45643:role/aliyunlogdefaultrole. For information about how to obtain the ARN, see Obtain the ARN of a RAM role.		
Shipping Size	The maximum size of raw log data that can be shipped to the OSS bucket in a shipping task. Valid values: 5 to 256. Unit: MB. If the size of shipped data exceeds the specified value, a new shipping task is automatically created.		
Compress	 Specifies whether to compress log data that is shipped to OSS. Valid values: No Compress: The log data that is shipped to OSS is not compressed. Compress (snappy): The snappy utility is used to compress the log data that is shipped to OSS. This way, the log data occupies less storage space of the OSS bucket. 		
Storage Format	The storage format of the log data that is shipped to OSS. Valid values: JSON, CSV, and Parquet. For more information, see Storage Formats.		
Ship Tags	Specifies whether to ship log tags.		
Shipping Time	The time period during which a shipping task runs. Valid values: 300 to 900. Default value: 300. Unit: seconds. If the specified time period expires, another shipping task is created.		

6. Click **OK**.

? Note

- After you configure a shipping rule, multiple shipping tasks can concurrently run. If the size of the data shipped from a shard reaches the specified threshold or the specified time period expires, another task is created.
- After you create a shipping task, you can check whether the shipping rule satisfies your business requirements based on the task status and the data shipped to OSS.

View OSS data

After log data is shipped to OSS, you can access the log data in the OSS console, or by using the OSS API, an SDK, or another method. For more information, see the **Objects > Search for objects** section of the **Service User Guide -Ojbect Storage Service(OSS)**.

The following script shows a sample OSS directory:

oss://OSS-BUCKET/OSS-PREFIX/PARTITION-FORMAT_RANDOM-ID

OSS-BUCKET is the name of the OSS bucket. OSS-PREFIX is the prefix of the directory in the OSS bucket. PARTITION-FORMAT is the partition format of the directory for a shipping task. The partition format is calculated based on the time when the shipping task is created. For more information, see strptime API. RANDOM-ID is the unique identifier of the shipping task.

Note The directory in the OSS bucket is created based on the time when the shipping task is created. For example, the shipping task is created at 00:00:00 on June 23, 2016 to ship data to OSS. The data is written to Log Service after 23:55:00 on June 22, 2016. The shipping interval is 5 minutes. To retrieve all logs shipped on June 22, 2016, you must check all objects in the 2016/06/22 directory. You must also check the 2016/06/23/00/ directory for the objects that are generated in the first 10 minutes of June 23, 2020.

Partition format

For each shipping task, log data is written to a directory of an OSS bucket. The directory is in the oss://OSS-BUCKET/OSS-PREFIX/PARTITION-FORMAT_RANDOM-ID format. A partition format is obtained by formatting the time when a shipping task is created. The following table describes the partition formats and directories that are obtained when a shipping task is created at 19:50:43 on January 20, 2017.

OSS Bucket	OSS Prefix	Partition format	OSS directory
test-bucket	test-table	%Y/%m/%d/%H/%M	oss://test-bucket/test- table/2017/01/20/19/50_1484913043351525 351_2850008
test-bucket	log_ship_oss_example	year=%Y/mon=%m/day=%d/log_%H%M%s	oss://test- bucket/log_ship_oss_example/year=2017/mo n=01/day=20/log_195043_14849130433515 25351_2850008.parquet
test-bucket	log_ship_oss_example	ds=%Y%m%d/%H	oss://test- bucket/log_ship_oss_example/ds=20170120/ 19_1484913043351525351_2850008.snappy

test-bucket	log_ship_oss_example	%Y%m%d/	oss://test- bucket/log_ship_oss_example/20170120/_148 4913043351525351_2850008 Note This format may prevent platforms such as Hive from parsing the log data in the OSS bucket. We recommend that you do not use this format.
test-bucket	log_ship_oss_example	%Y%m%d%H	oss://test- bucket/log_ship_oss_example/2017012019_1 484913043351525351_2850008

You can use Hive, MaxCompute, or Data Lake Analytics (DLA) to analyze OSS data. In this case, if you want to use partition information, you can set PARTITION-FORMAT in the key=value format. For example, you can set the partition format to oss://testbucket/log_ship_oss_example/year=2017/mon=01/day=20/log_195043_1484913043351525351_2850008.parquet. In this example, year, mon, and day are specified as three partition keys.

What to do next

After shipping tasks are created based on a shipping rule, you can modify the shipping rule. You can also disable the data shipping feature, view the statuses and error messages of the tasks, and retry failed tasks on the **OSS Shipper** page of a Logstore.

- Modify the shipping rule.
- Click Settings to modify the shipping rule. For information about the parameters, see Procedure.
- Disable the data shipping feature.
- Click **Disable**. The data in the Logstore is no longer shipped to OSS.
- View the statuses and error messages of the tasks.

You can view the log shipping tasks of the last two days and their statuses.

Statuses of a shipping task

Status	Equivalent
Succeeded	The shipping task has succeeded.
Running	The shipping task is running. Check whether the task succeeds later.
Failed	The shipping task has failed. If the task cannot be restarted due to external causes, troubleshoot the failure based on the error message and retry the task.

Error messages

If a shipping task fails, an error message is returned for the task.

Error message	Error cause	Solution
UnAuthorized	The error message returned because the AliyunLogDefaultRole role does not have the required permissions.	 To fix the error, check the following configurations: Check whether the AliyunLogDefaultRole role is created by the OSS bucket owner. Check whether the specified ID of the Alibaba Cloud account in the permission policy is valid. Check whether the AliyunLogDefaultRole role is granted the write permissions on the OSS bucket. Check whether the ARN of the AliyunLogDefaultRole role that you entered in the RAM Role field is valid.
ConfigNotExist	The error message returned because the task does not exist.	Check whether the data shipping feature is disabled. If the feature is disabled, enable the feature, configure a shipping rule, and then retry the shipping task.
InvalidOssBucket	The error message returned because the specified OSS bucket does not exist.	To fix the error, check the following configurations:Check whether the OSS bucket resides in the same region as the Log Service project.Check whether the specified bucket name is valid.
InternalServerError	The error message returned because an internal error has occurred in Log Service.	Retry the failed shipping task.

Retry a shipping task

By default, if a shipping task fails, Log Service retries the task based on the retry policy. You can also manually retry the task. By default, Log Service retries all tasks of the last two days. The minimum interval between two consecutive retries is 15 minutes. If a task fails for the first time, Log Service retries the task 15 minutes later. If the task fails for the second time, Log Service retries the task 30 minutes later. If the task fails for the second time, Log Service retries the task 30 minutes later. If the task fails for the third time, Log Service retries the task 60 minutes later. A similar method is used for subsequent attempts.

To immediately retry a failed task, you can click **Retry All Failed Tasks** or **Retry** on the right of the task. You can also use the Log Service API or an SDK to retry a task.

4.8.1.3. Obtain the ARN of a RAM role

When you use a RAM user to ship data from Log Service to Object Storage Service (OSS), you must first create a Resource Access Management (RAM) role and specify the ARN of the RAM role. This topic describes how to create a RAM role and obtain the ARN of a RAM role.

Procedure

- 1. Log on to the Log Service console
- 2. In the top navigation bar, click **Configurations**.
- 3. On the Service-Linked Roles page, click Create Service-linked Role.

- In the Service Name drop-down list, select Log Service, in the Organization Name drop-down list, select the organization that you created. and click OK.
- 5. On the RAM Service Role page, enter AliyunLogDefaultRole in the Role Name search box and click Search.
- 6. Find the organization that you created, obtain the ARN of the RAM role.

In the search results, the value in the Role Identifier column is the ARN of the RAM role.

Role Name	Organization Name	Role Identifier		Service Name	Description		Created At
AliyunLogDefaultRole	baseservicetest	acs:ram::10	gdefaultrole	Log Service	Log Service will use this role to acce s.	ss your resources in other service	Aug 21, 2023

4.8.1.4. Storage Formats

Different storage formats are supported when Log Service ships logs to OSS, including JSON, CSV, and Parquet. This topic describes the field details of the formats.

JSON format

You can set the storage format for the data that is shipped to OSS. The following table shows how to set the **storage format** to **JSON**. For more information, see Configure a data shipping rule.

Compression type	File extension	Example file address	Description
Uncompressed	N/A	oss://oss-shipper- shenzhen/ecs_test/2016/01/26/20/54_145381289305 9571256_937	You can download the raw JSON object to the local host and open each object as a text file. The following example is the content of a sample file: {"time":1453809242,"topic":"","sou rce":"10.170.****","ip":"10.200.**.***" ,"time":"26/Jan/2016:19:54:02 +0800","url":"POST
snappy	.snappy	oss://oss-shipper- shenzhen/ecs_test/2016/01/26/20/54_145381289305 9571256_937.snappy	JSON objects are compressed by using Snappy. For more information, see Decompression tools for Snappy-compressed files.

CSV-format

You can set the storage format for the data that is shipped to OSS. The following table shows how to set the **storage format** to **CSV**. For more information, see Configure a data shipping rule.

The following table describes the parameters. For more information, see Common Format and MIME Type for Comma-Separated Values (CSV) Files and PostgreSQL 9.4.26 Documentation.

Parameter	Description		
CSV Fields	The names of the log fields that you want to ship to OSS. You can view log fields on the Raw Logs tab of a Logstore and enter the names of the fields that you want to ship to OSS in the Key Name column. The log fields that you can ship to OSS include the fields in the log content and the reserved fields such as _time_, _topic_, and _source ⑦ Note The keys that you enter in the CSV Fields section must be unique.		
Delimiter	You can use commas (,), vertical bars (), spaces, or tabs to delimit fields.		
Escape Character	If a field contains a delimiter, you must use an escape character to enclose the field. This ensures that the field is not delimited.		
Invalid Fields	If a key that you specify in the CSV Fields section does not exist, enter the value of the key in the Invalid Fields field.		
Shipped Fields	If you turn on the Shipped Fields switch, field names are written in a CSV file.		

The following table lists the directories in OSS buckets that store the data shipped from Log Service.

Compression type	File extension	Example	Description
No	.CSV	oss://oss-shipper- shenzhen/ecs_test/2016/01/26/20/54_145381289 3059571256_937.csv	You can download the raw JSON object to the local host and open the object as a text file.
snappy	.snappy.csv	oss://oss-shipper- shenzhen/ecs_test/2016/01/26/20/54_145381289 3059571256_937.snappy.csv	Decompression tools for Snappy compressed files For more information, see Decompression tools for Snappy-compressed files.

Parquet-format

You can set the storage format for the data that is shipped to OSS. The following figure shows how to set the **storage format** to **Parquet**. For more information, see Configure a data shipping rule.

The following table describes the related parameters.

Parameter

Description

Key Name	 The name of the log field that you want to ship to OSS. You can view log fields on thRaw Logs tab of a Logstore. You can also enter the names of the fields that you want to ship to OSS in the Key Name column. When the fields are shipped to OSS, they are stored in the Parquet format in the order that the field names are entered. The names of the fields are the column names in OSS. The log fields that you can ship to OSS include the fields in the log content and the reserved fields such as <u>_time_</u>, <u>topic_</u>, and <u>_source_</u>. The value of a field in the Parquet format is null in the following two scenarios: The field does not exist in logs. The value of the field fails to be converted from the string type to a non-string type, for example, double or Int64. 	
	⑦ Note The keys that you enter in the Parquet Keys field must be unique.	
	The Parquet storage format supports six data types: string, Boolean, Int32, Int64, float, and double.	
Туре	Log fields are converted from the string type to a data type that the Parquet storage format supports. If the data type of a log field fails to be converted, the value of the log field is null.	

The following table lists the directories in OSS buckets that store data shipped from Log Service.

Compression type	File extension	Example	Description
Uncompressed	.parquet	oss://oss-shipper- shenzhen/ecs_test/2016/01/26/20/54_145381289 3059571256_937.parquet	You can download the OSS buckets to the local host and use the parquet-tools utility to open the objects. For more information about the parquet-tools utility, visit parquet-tools.
Snappy	.snappy.parquet	oss://oss-shipper- shenzhen/ecs_test/2016/01/26/20/54_145381289 3059571256_937.snappy.parquet	You can download the OSS buckets to the local host and use the parquet-tools utility to open the objects. For more information about the parquet-tools utility, visit parquet-tools.

4.8.1.5. Decompress Snappy compressed files

When you ship data from Log Service to Object Storage Service (OSS), you can use Snappy to compress OSS objects. After the data is shipped to OSS, you can decompress OSS objects by using the C++ library, Java library, Python library, and decompression tool for Linux.

Use the C++ library to decompress OSS objects

Download the C++ library from the snappy page and use the Snappy.Uncompress method to decompress Snappy compressed OSS objects.

Use the Java library to decompress OSS objects

Download the Java library from the xerial snappy-java page and use the Snappy.Uncompress or Snappy.SnappyInputStream method to decompress Snappy compressed OSS objects. The SnappyFramedInputStream method is not supported.

Note If you use Java Library 1.1.2.1, some Snappy compressed OSS objects may fail to be decompressed. For more information, see Bad handling of the MAGIC HEADER. To fix this issue, you can use Java Library 1.1.2.6 or later.

<dependency> <groupId>org.xerial.snappy</groupId> <artifactId>snappy-java</artifactId> <version>1.0.4.1</version> <type>jar</type> <scope>compile</scope> </dependency>

Snappy.Uncompress

String fileName = "C:\\Downloads\\36_1474212963188600684_4451886.snappy"; RandomAccessFile randomFile = new RandomAccessFile(fileName, "r"); int fileLength = (int) randomFile.length(); randomFile.seek(0); byte[] bytes = new byte[fileLength]; int byteread = randomFile.read(bytes); System.out.println(fileLength); System.out.println(byteread); byte[] uncompressed = Snappy.uncompress(bytes); String result = new String(uncompressed, "UTF-8"); System.out.println(result);

Snappy.SnappyInputStream

```
String fileName = "C:\\Downloads\\36_1474212963188600684_4451886.snappy";
SnappyInputStream sis = new SnappyInputStream(new FileInputStream(fileName));
byte[] buffer = new byte[4096];
int len = 0;
while ((len = sis.read(buffer)) != -1) {
    System.out.println(new String(buffer, 0, len));
}
```

Use the Python Library to decompress OSS objects

1. Download and install the Python library.

 Run the decompression script. The following example is a sample decompression script:

```
import snappy
compressed = open('/tmp/temp.snappy').read()
snappy.uncompress(compressed)
```

(2) Note The following two commands cannot be used to decompress Snappy compressed OSS objects. These commands can be used only in Hadoop mode (hadoop_stream_decompress) or streaming mode (stream_decompress).

python -m snappy -c uncompressed_file compressed_file.snappy python -m snappy -d compressed_file.snappy uncompressed_file

Use decompression tools for Linux to decompress OSS buckets

Log Service allows you to decompress Snappy compressed files by using the decompression tool for Linux. Click snappy_tool to download the tool. Replace 03_1453457006548078722_44148.snappy and 03_1453457006548078722_44148 in the following code with the values specific to your environment and then run the following code:

./snappy_tool 03_1453457006548078722_44148.snappy 03_1453457006548078722_44148 compressed.size: 2217186 snappy::Uncompress return: 1 uncompressed.size: 25223660

4.9. Log applications

4.9.1. Trace

4.9.1.1. Usage notes

Log Service provides the Trace application based on the native OpenTelemetry protocol to implement distributed tracing. You can use the application to import, store, analyze, and visualize trace data. You can also use the application to manage trace data based on AlOps. This topic describes the background information, features, and assets of the Trace application.

Background information

In modern IT systems, such as cloud-native systems and microservice systems, an external request often requires multiple internal services, middleware, and machines to call each other. During the call process, various issues may occur and cause external service failures or an increased latency. This affects user experience. To identify and analyze issues, you can use the distributed tracing method.

Distributed tracing can provide information about call relationships, latencies, and results of an entire service call link. Distributed tracing is suitable for systems that require interaction between multiple services, such as cloud-native, distributed, and microservices systems.

OpenTelemetry is a globally accepted standard of distributed tracing, and is compatible with OpenTracing and OpenCensus clients. OpenTelemetry provides a collection of APIs, SDKs, and tools. You can use OpenTelemetry to instrument, generate, collect, and export various observable data, including traces, logs, and metrics.

Trace application of Log Service

OpenTelemetry defines data formats, and generates, collects, and sends data. OpenTelemetry does not analyze or visualize data. The Trace application is implemented based on the OpenTelemetry protocol. You can use the application to collect trace data from OpenTelemetry and other platforms, such as Jaeger, Zipkin, and SkyWalking. You can also store, analyze, and visualize trace data.



- · Multiple import methods
 - $\circ~$ You can import trace data over multiple protocols such as OpenTelemetry, Jaeger, and Zipkin.
- $\circ~$ You can import trace data in more than 10 programming languages
- $\circ~$ You can import trace data from multiple trace platforms.
- You can import trace data over the Internet or an Alibaba Cloud internal network. An Alibaba Cloud internal network can be the classic network or a virtual private cloud (VPC).

• Compliance with OpenTelemetry Trace 1.0

The trace data format of Log Service complies with OpenTelemetry Trace 1.0 and meets the format requirements of cloud-native systems and microservice systems for trace data.

High performance

You can import petabytes of data per day, extract and analyze metrics, precompute data, and sample 100% of trace data in large-scale scenarios.

Scalability

- You can configure custom data retention periods. Log Service can dynamically scale the capacity of Logstores to meet your business requirements. • Various trace-related features
- You can view trace and service details, query and analyze trace data, analyze dependencies, and perform custom SQL analysis.
- High compatibility with downstream applications

Trace data and calculated metrics in Log Service are compatible with various stream processing platforms and batch computing engines. The Trace application also supports the custom processing of subscription data.

Multiple built-in AlOps algorithms
 The Trace application can automatically analyze the impact of trace data on performance and error rates. This helps developers identify the root causes of various issues in complex scenarios.

Assets

All assets that are created by the Trace application are stored in a specified project. The project contains the following assets:

Logstore

() Important Do not update or delete indexes in the following Logstores. Otherwise, the Trace application becomes unavailable.

- {instance}-traces: stores the raw trace data that is uploaded.
- {instance}-traces-metrics: stores the intermediate results of aggregated metrics after trace data is calculated.
- $\circ~$ {instance}-traces-deps: stores the data of dimension call relationships after trace data is calculated.
- {instance}-logs: stores the raw log data that is uploaded.
- Metricstore

{instance}-metrics: stores the uploaded metrics.

- Scheduled SQL
 - {instance}-metric_info: queries the metrics that are used to aggregate trace data.
- {instance}-service: queries the dependencies that are used to aggregate trace data at the service granularity.
- {instance}-service_name_host: queries the dependencies that are used to aggregate trace data at the service, name, and host granularities.
- {instance}-service_name_host_resource: queries the dependencies that are used to aggregate trace data at the service, name, host, and resource granularities.
- Dashboard
 - Import overview: displays the basic information about imported trace data, such as the number of traces, number of spans, and import status of each service.
- Statistics: displays the statistics of imported trace data, such as the latency, queries per second (QPS), and error rate.

4.9.1.2. Trace data formats

This topic describes the trace data formats that are supported by Log Service.

Log Service is compatible with the trace data formats that are defined in OpenTelemetry Trace 1.0. If trace data is written by using protocols such as OpenTelemetry, Jaeger, Zipkin, OpenCensus, and SkyWalking, Log Service automatically maps the trace data to a data format defined in OpenTelemetry. For other trace data, you can use the data transformation feature to map the data to a data format supported by Log Service.

Field	Туре	Required	Description	Example
host	String	No	The hostname of the host where the resource resides. The value is extracted from host.name of the resource field.	test-host
service	String	Yes	The service name of the resource. The value is extracted from service.name of the resource field.	test-service
resource	JSON Object	No	Other resource-related fields. The host and service fields are also resource-related fields. Examples: process ID, process name, and pod name. For more information, see Resource Semantic Conventions .	{"k8s.pod.name":"xxxx", "k8s.pod.namespace":"k ube-system"}
otlp.name	String	No	The name of Trace SDK.	go-sdk
otlp.version	String	No	The version of Trace SDK.	v1.0.0
name	String	Yes	The name of the span.	/get/314159
kind	String	No	The type of the span. Examples: CLIENT and SERVER. For more information, see SpanKind.	SERVER
traceID	String	Yes	The ID of the trace. The value is a hexadecimal string.	0123456789abcde01234 56789abcde
spanID	String	Yes	The ID of the span. The value is a hexadecimal string.	0123456789abcde
parentSpanID	String	Yes	The ID of the parent span. The value is a hexadecimal string.	0123456789abcde
links	JSON Array	No	The other spans that are associated with the trace data. For more information, see Specifying links.	[{"TracelD": "abc", "Spanld": "abc", "TraceState": "", "Attributes": { "k": "v" } }]
logs	JSON Array	No	The log and event information that is associated with the trace data. For more information, see Add Events.	None

traceState	String	No	The trace state, which is defined in the World Wide Web Consortium (W3C) specification. For more information, see W3C Trace Context Specification.	None
start	INT	Yes	The start time. The value is a UNIX timestamp. Unit: microseconds.	1615882567123456
end	INT	No	The end time. The value is a UNIX timestamp. Unit: microseconds.	1615882567234567
duration	INT	Yes	The latency. The value is the difference between the start and end fields. Unit: microseconds.	1020
attribute	JSON Object	Yes	The attribute information about the span, such as the URL and status code of the HTTP request. For more information, see Attribute Naming.	{"custom":"custom","ho st.hostname":"myhost"," my-label":"myapp- type","null- value":"","service.name" :"myapp"}
statusCode	String	Yes	The status code of the HTTP request. Valid values: OK, ERROR, and UNSET. The value UNSET is equivalent to the value OK.	ERROR
statusMessage	String	No	The status message of the HTTP request.	stack overflow

4.9.1.3. Create a trace instance

A trace instance of Log Service is used to manage all collected trace data. You can query and analyze trace data, and view the details of trace data. You can also view service metrics. This topic describes how to create a trace instance in the Log Service console.

Procedure

- 1. Log on to the Log Service console.
- 2. On the Intelligent O&M tab in the Log Application section, click Trace.
- 3. On the page that appears, click Create Instance
- 4. In the Create Instance panel, configure the following parameters and click OK.

Parameter	Description
Name	The name of the trace instance.
Description	The description of the trace instance.
Project	The project that is used to store trace data. Select the project that you use to store your trace data from the drop-down list. If no project is available, click Create Now to create a project. For more information, seeCreate a project.
Instance ID	The ID of the trace instance. The instance ID can contain lowercase letters, digits, and hyphens (-). It must start with lowercase letters or digits, and cannot exceed 20 characters in length.

After the instance is created, it enters the Running state. You can view the basic information of the instance in the instance list.

What to do next

Import trace data

4.9.1.4. Import trace data

4.9.1.4.1. Overview

You can import cloud native trace data from OpenTelemetry to Log Service. You can also import trace data from other tracing systems to Log Service. This topic describes the methods that can be used to connect to the Trace application of Log Service.

Import methods

Log Service supports the following methods to import trace data:

- Use OpenTelemetry, Jaeger, Zipkin, and OpenCensus to import trace data to Log Service. If you import trace data from Jaeger to Log Service, you can use only an HTTPS or gRPC method.
- Use the OpenTelemetry Collector to forward trace data from OpenTelemetry, Jaeger, Zipkin, OpenCensus, AWS X-Ray, and Splunk SignalFX to Log Service. In this method, all protocols are supported for Jaeger.
- Use Logtail to forward trace data from SkyWalking to Log Service.
- Use a custom protocol to import trace data to Log Service. Then, you can convert the format of the trace data to the OpenTelemetry format by using the data transformation feature of Log Service.

Instructions on selecting import methods

Before you select an import method to import trace data to Log Service, take note of the following instructions:

- Use OpenTelemetry to import trace data.
 The OpenTelemetry protocol is a globally recognized standard that is used to import trace data. To connect with all required components, multiple open source software applications comply with the OpenTelemetry protocol.
- Comply with the OpenTracing or OpenTelemetry protocol to connect with other open source systems.
- If you do not use an open source standard protocol, we recommend that you use the same import method for all services from which trace data is imported in your tracing system. Otherwise, the trace data that is collected may be incomplete.



Details of import methods

Log Service supports multiple import methods that have different automation levels of instrumentation and import complexity. Import methods are listed for common tracing platforms, such as OpenTelemetry, SkyWalking, Jaeger, and Zipkin.

Import methods for trace data in different programming languages

You can import trace data to Log Service by using automatic instrumentation or semi-automatic instrumentation.

- Automatic instrumentation: Developers do not need to modify frameworks or code. Tracing systems automatically set up instrumentation.
- Semi-automatic instrumentation: Developers need to manually install dependencies or modify code.

Language	Import method	Automation level	Complexity
	Import trace data by using OpenTelemetry	Automatic	Low
Java	Forward trace data by using the OpenTelemetry Collector	Automatic	Medium
	Import trace data from Apache SkyWalking to Log Service	Automatic	Medium
Golang	Import trace data by using OpenTelemetry	Semi-automatic	Low
Golding	Forward trace data by using the OpenTelemetry Collector	Semi-automatic	Low
Python	Import trace data by using OpenTelemetry	Semi-automatic	Medium
	Forward trace data by using the OpenTelemetry Collector	Semi-automatic	Medium
NodelS	Import trace data by using OpenTelemetry	Semi-automatic	Medium
Nodejs	Forward trace data by using the OpenTelemetry Collector	Semi-automatic	Medium
РНР	Import trace data by using Zipkin	Manual	High
C++	Import trace data by using Jaeger	Manual	High
	Import trace data by using OpenTelemetry	Semi-automatic	Medium
C#	Forward trace data by using the OpenTelemetry Collector	Semi-automatic	Medium

Cloud Defined Storage

User Guide-Log Service

	Import trace data from Apache SkyWalking to Log Service	Automatic	Medium
Duct	Import trace data by using OpenTelemetry	Manual	High
KUSL	Forward trace data by using the OpenTelemetry Collector	Manual	High
Duby	Import trace data by using OpenTelemetry	Manual	High
Kuby	Forward trace data by using the OpenTelemetry Collector	Manual	High

Import methods for trace data from different platforms

Tracing platform	Import method	Complexity
	Import trace data from OpenTelemetry	Low
OpenTelemetry	Forward trace data by using the OpenTelemetry Collector	Medium
	Import trace data from Jaeger	Low
Jaeger	Forward trace data to ARMS by using OpenTelemetry Collector	Medium
	Import trace data from Zipkin	Low
Zipkin	Use the OpenTelemetry Collector to forward trace data	Medium
SkyWalking	Forward trace data by using Logtail	Medium
OpenCensus	Forward trace data by using the OpenTelemetry Collector	Medium
AWS X-Ray	Forward trace data by using the OpenTelemetry Collector	High
Splunk SignalFX	Forward trace data by using the OpenTelemetry Collector	High

Scenarios

· Build a tracing system

If your system is connected to the Trace application for the first time, we recommend that you use OpenTelemetry to upload your trace data to Log Service. However, the related programming language may not support OpenTelemetry import methods or the import methods may not meet your requirements. In this case, you can use the import methods that support the OpenTracing or OpenCensus protocol to import data from Jaeger or Zipkin.

- Upgrade an existing tracing system
 - If your current system uses a tracing service, you can select an import method based on the actual scenario.
 - The tracing system is stably running.
 - If trace data can be uploaded to the OpenTelemetry Collector in the tracing system, you can use the OpenTelemetry Collector to forward the trace data to Log Service.
 - If the tracing system uses a custom protocol or another protocol that is not the OpenTelemetry or OpenTracing protocol, you can print trace data to a file. Then, you can upload the file to Log Service by using Logtail, and use the data transformation feature to convert the data format to the OpenTelemetry format.
 - The tracing system does not meet your business requirements or the tracing system needs to be upgraded.
 - If the tracing system uses the OpenTracing or OpenCensus protocol, you can smooth and migrate trace data. In this case, you must upload trace
 data from the original system to the OpenTelemetry Collector. Then, forward the trace data to Log Service. During this process, the original
 protocol is replaced by the OpenTelemetry protocol. Then, the OpenTelemetry protocol is used to import the trace data to Log Service.
 - If the tracing system uses another protocol, you must replace the protocol. Otherwise, trace data may be incomplete during the replacement process.
- Deploy on-premises tracing systems

If you deploy your business applications in a data center and only some gateways connect to the Internet or Express Connect circuits, you can deploy the OpenTelemetry Collector on these gateways. Then, you can send trace data from other machines to the gateways, and then forward the trace data to Log Service by using the OpenTelemetry Collector.

Trace Demo

Log Service provides demos that demonstrate how to import trace data for different programming languages. For more information, see Trace Demos.

4.9.1.4.2. New import methods

4.9.1.4.2.1. Import trace data from Java applications to Log Service by using

OpenTelemetry SDK for Java

This topic describes how to import trace data from Java applications to Log Service by using OpenTelemetry SDK for Java.

Prerequisites

- A trace instance is created. For more information, see Create a trace instance.
- A Java development environment is set up. The Java version is 8 or later.

(2) Note We recommend that you use Java Development Kit (JDK) 8u252 or a later version.

Method 1: (Recommended) Use a Java agent to automatically upload trace data

You can use a Java agent to automatically upload trace data to Log Service in dozens of Java frameworks. For more information, see Supported libraries, frameworks, application servers, and JVMs.

() Important You cannot use a Java agent together with a SkyWalking agent or a Zipkin agent. If you use a Java agent together with a SkyWalking agent or a Zipkin agent, undefined behavior may occur.

1. Download the latest version of a Java agent.

2. Configure the Java agent.

The following code provides an example on how to configure the environment variables for the -javaagent parameter of a Java Virtual Machine (JVM). For more information, see opentelemetry-java-instrumentation. You must replace the variables such as *\${endpoint}* and *\${project}* in the code with the actual values.

export OTEL_EXPORTER_OTLP_PROTOCOL=grpc

export OTEL_EXPORTER_OTLP_ENDPOINT=https://\${endpoint}

export OTEL_EXPORTER_OTLP_COMPRESSION=gzip

export OTEL_EXPORTER_OTLP_HEADERS=x-sls-otel-project=\${project},x-sls-otel-instance-id=\${instance},x-sls-otel-ak-id=\${access-key-id},x-sls-ote

java -javaagent:/path/to/opentelemetry-javaagent-all.jar -

Dotel.resource.attributes=service.namespace=\${service.namespace}, service.name=\${service}, service.version=\${version}, host.name=\${host}, deployment vironment= \${environment} -jar /path/to/your/app.jar

Table 1. Variables

Variable	Description	Example
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. The fixed port number is 10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations.	trace.test-region-id.sls-pub.test- internet-domain:10010
<i>\${project}</i>	The name of the Log Service project.	test-project
<i>\${instance}</i>	The ID of the trace instance. For more information, seeCreate a trace instance.	test-traces
\${access-key-id}	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 	None
\${access-key-secret}	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None
<pre>\${service.namespace}</pre>	The namespace to which the service belongs.	order
<i>\${service}</i>	The name of the service. Specify the value based on your business requirements.	payment
\${version}	The version of the service. We recommend that you specify a version in the $\ensuremath{\textbf{va.b.c}}$ format.	v0.1.2
\${host}	The hostname.	localhost
\${environment}	The deployment environment. Example: test environment or production environment. Specify the value based on your business requirements.	pre

Method 2: Manually construct and upload trace data

If you use a self-managed framework or have special requirements, you can manually construct trace data and upload the data to Log Service. In this example, Maven is used to construct trace data. For more information, see OpenTelemetry QuickStart.

1. Add Maven dependencies.

<dependency></dependency>
<groupid>io.opentelemetry</groupid>
<artifactid>opentelemetry-sdk</artifactid>
<version>1.9.0</version>
<dependency></dependency>
<groupid>io.opentelemetry</groupid>
<artifactid>opentelemetry-exporter-otlp</artifactid>
<version>1.9.0</version>
<dependency></dependency>
<groupid>io.grpc</groupid>
<artifactid>grpc-netty-shaded</artifactid>
<version>1.41.0</version>
<dependency></dependency>
<groupid>io.opentelemetry</groupid>
<artifactid>opentelemetry-semconv</artifactid>
<version>1.9.0-alpha</version>
<scope>runtime</scope>

2. Add initialization code.

```
You must replace the variables such as ${endpoint} and ${project} in the following code with the actual values. For more information about the
variables, see Variable
 OtlpGrpcSpanExporter grpcSpanExporter = OtlpGrpcSpanExporter.builder()
             // When you set the .setEndpoint parameter, add https://. Example: https://test-project.cn-hangzhou.log.aliyuncs.com:10010.
             .setEndpoint("https://${endpoint}:10010")
             .addHeader("x-sls-otel-project", "${project}")
             .addHeader("x-sls-otel-instance-id", "${instance}")
             .addHeader("x-sls-otel-ak-id", "${access-key-id}")
              .addHeader("x-sls-otel-ak-secret", "${access-key-secret}")
             .build();
         SdkTracerProvider tracerProvider = SdkTracerProvider.builder()
             .addSpanProcessor(BatchSpanProcessor.builder(grpcSpanExporter).build())
             .setResource (Resource.create (Attributes.builder()
                 .put(ResourceAttributes.SERVICE_NAME, "${service}")
                 .put (ResourceAttributes.SERVICE NAMESPACE, "${service.namespace}")
                 .put(ResourceAttributes.SERVICE_VERSION, "${version}")
                 .put(ResourceAttributes.HOST_NAME, "${host}")
                  .build()))
             .build();
         OpenTelemetry openTelemetry = OpenTelemetrySdk.builder()
             .setTracerProvider(tracerProvider)
             .setPropagators(ContextPropagators.create(W3CTraceContextPropagator.getInstance())))
             .build();
         Tracer tracer =
             openTelemetry.getTracer("instrumentation-library-name", "1.0.0");
         Span parentSpan = tracer.spanBuilder("parent").startSpan();
         try {
             Span childSpan = tracer.spanBuilder("child")
                 .setParent(Context.current().with(parentSpan))
                 .startSpan();
             childSpan.setAttribute("test", "vllelel");
             // do stuff
             childSpan.end();
         } finally {
            parentSpan.end();
```

FAQ

What do I do if the "Could not find TLS ALPN provider" error message is returned by the Java agent when the OpenJDK version is earlier than 8u252?

```
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at sun.instrument.InstrumentationImpl.loadClassAndStartAgent/InstrumentationImpl.java:430)
at sun.instrument.InstrumentationImpl.loadClassAndStartAgent/InstrumentationImpl.java:430)
Caused by: java.lang.IllegalStateException: Could not find TLS ALPN provider; no working netty-tenative, Conscrypt, or Jetty NPN/ALPN available
at io.grpc.netty.GrpcSslContexts.defaultSslProvider(GrocSslContexts.java:241)
at io.grpc.netty.GrpcSslContexts.forClient(GrpcSslContexts.java:241)
at io.grpc.netty.GrpcSslContexts.forClient(GrpcSslContexts.java:94)
```

To resolve this issue, perform the following steps:

- 1. Download a package.
- 2. Run the following command to add the required JAR files.

\${youpath} specifies the path to each JAR file. Replace each \${youpath} variable with the actual value.

java -Xbootclasspath/p:\${youpath}/netty-tcnative-boringssl-static-2.0.25.Final.jar -javaagent:\${youpath}/opentelemetry-javaagent-all.jar -ja r \${youpath}/demo2-0.0.1-SNAPSHOT.jar

What to do next

- View the details of a trace instance
- Ouerv and analyze trace data

4.9.1.4.2.2. Import trace data from Golang applications to Log Service by using

OpenTelemetry SDK for Go

This topic describes how to import trace data from Golang applications to Log Service by using OpenTelemetry SDK for Go.

Prerequisites

- A trace instance is created. For more information, see Create a trace instance.
- A Golang development environment is set up. The Go version is 1.13 or later.

Procedure

- 1. Initialize an OpenTelemetry provider.
- 2. Check whether the conditions for importing data in semi-automatic mode are met.
 - If the conditions are met, you can import trace data in semi-automatic mode.
 - If the semi-automatic mode does not meet your requirements, you must manually import the trace data.
 - · If the conditions are not met, you can import trace data in manual mode

Step 1: Initialize an OpenTelemetry provider

Log Service offers a provider that allows you to build dependencies and upload the dependencies to Log Service. This provider helps simplify the use of

an OpenTelemetry provider. For more information, see opentelemetry-go-provider-sls. () Important You must initialize an OpenTelemetry provider before you create traces and register metrics. You can run code or configure environment variables to initialize an OpenTelemetry provider. • Run code to initialize an OpenTelemetry provider. i. Add dependencies. module opentelemetry-golang-sample go 1.13 require (github.com/aliyun-sls/opentelemetry-go-provider-sls v0.8.0 go.opentelemetry.io/otel v1.11.2 go.opentelemetry.io/otel/metric v0.34.0 go.opentelemetry.io/otel/trace v1.11.2) ii. Write initialization code. Replace the variables in the following code with the actual values. For more information about the variables, see Variables. package main import ("github.com/aliyun-sls/opentelemetry-go-provider-sls/provider") func main() { slsConfig, err := provider.NewConfig(provider.WithServiceName("\${service}"), provider.WithServiceNamespace("\${service.namespace}"), provider.WithServiceVersion("\${version}"), provider.WithTraceExporterEndpoint("\${endpoint}"), provider.WithMetricExporterEndpoint("\${endpoint}"), provider.WithSLSConfig("S[mpo]ect)", "\$(access-key-id)", "\$(access-key-secret)"))
// Invoke the panic() function. If the initialization fails, the OpenTelemetry provider exits. You can also use other error handling m ethods. if err != nil { panic(err) if err := provider.Start(slsConfig); err != nil { panic(err) defer provider.Shutdown(slsConfig) // Add business logic code. } Table 1. Variables Variable Description Example

Vallable	beschption	Example
\${service}	The name of the service. Specify the value based on your business requirements.	payment
\${service.namespace}	The namespace to which the service belongs.	order
\${version}	The version of the service. We recommend that you specify a version in the $\ensuremath{\textbf{va.b.c}}$ format.	v0.1.2
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. The fixed port number is 10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations.	trace.test-region-id.sls-pub.test-inter- domain:10010
<i>\${project}</i>	The name of the Log Service project.	test-project
<i>\${instance}</i>	The ID of the trace instance. For more information, seeCreate a trace instance.	test-traces
\${access-key-id}	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 	None
\${access-key-secret}	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None

• Configure environment variables to initialize an OpenTelemetry provider.

Configuration method	Environment variable	Required	Description	Default value
----------------------	----------------------	----------	-------------	---------------

Cloud Defined Storage

User Guide-Log Service

WithServiceName	SLS_OTEL_SERVICE_NA ME	Yes	The name of the service. Specify the value based on your business requirements.	None
WithServiceNamespace	SLS_OTEL_SERVICE- NAMESPACE	No	The namespace to which the service belongs.	order
WithServiceVersion	SLS_OTEL_SERVICE_VER SION	Yes	The version of the service. We recommend that you specify a version in the va.b.c format.	v0.1.0
WithSLSConfig	SLS_OTEL_PROJECT, SLS_OTEL_INSTANCE_ID, SLS_OTEL_ACCESS_KEY_ ID, and SLS_OTEL_ACCESS_KEY_ SECRET	No	 The information about Log Service resources. The information includes the name of a project, name of a trace instance, AccessKey ID of an account that has only the write-permissions on the project, and AccessKey secret of the account. For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see the Obtain AccessKey credentials topic in Preparations. 	None
WithTraceExporterEndp oint	SLS_OTEL_TRACE_ENDP OINT	No	The endpoint of the Log Service project. Format: data.\${region_id}.sls-pub.\${internet- domain}:10010. The fixed port number is 10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations. ⑦ Note • If you set the variable tostdout, data is printed to standard output. • If you leave the variable empty, trace data is not uploaded to Log Service.	stdout
WithTraceExporterInsec ure	SLS_OTEL_TRACE_INSEC URE	No	 Specifies whether to transfer data by using a method that is not secure. Valid values: true false Note If you want to directly transfer data to Log Service, you must set the variable to false. 	false
WithMetricExporterEndp oint	SLS_OTEL_METRIC_ENDP OINT	No	The endpoint of the Log Service project. Format: data.\${region_id}.sls-pub.\${internet- domain}:10010. The fixed port number is 10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations. ⑦ Note • If you set the variable to stdout, data is printed to standard output. • If you leave the variable empty, metric data is not uploaded to Log Service.	stdout
WithMetricExporterInsec ure	SLS_OTEL_METRIC_INSE CURE	No	 Specifies whether to transfer data by using a method that is not secure. Valid values: true false Note If you want to directly transfer data to Log Service, you must set the variable to false. 	false
WithResourceAttributes	None	No	The additional tag information, such as the environment and zone.	None
WithResource	OTEL_RESOURCE_ATTRI BUTES	No	The additional tag information, such as the environment and zone. Format: key1=value1,key2=value2 .	None
WithMetricReportingPeri od	SLS_OTEL_METRIC_EXPO RT_PERIOD	No	The interval of reporting metric data. We recommend that you set the interval to a value from 15s to 60s.	30s
WithErrorHandler	None	No	The error handling function. If an internal SDK error occurs, the system calls this function. This function is equivalent to the WithErrorHandlerFunc function.	None
WithErrorHandlerFunc	None	No	The error handling function.	None
None	SLS_OTEL_ATTRIBUTES_ ENV_KEYS	No	The additional tag information, such as the environment and zone. This variable is similar to OTEL_RESOURCE_ATTRIBUTES. However, the values of attribute keys that are defined in the SLS_OTEL_ATTRIBUTES_ENV_KEYS variable are read from other environment variables. SLS_OTEL_ATTRIBUTES_ENV_KEYS is commonly used in Kubernetes clusters to pad some template values to specified environment variables. Format: env- key-1 env-key-2 env-key-3.	None

Step 2: Import data

· Semi-automatic mode: recommended OpenTelemetry provides automatic instrumentation solutions for various basic libraries. If your business rely on these libraries, you can use the automatic instrumentation solutions to import data. For more information about basic libraries, see Instrumentation • Use the .NET or HTTP framework to import data The following sample code is created based on go.opentelemetry.io/contrib/instrumentation/net/http/otelhttp v0.37.0 . For more information, see otel-http-example. Replace the variables in the following code with the actual values. For more information about the variables, see Variables. package main import ("fmt' "io" "net/http" "time" "github.com/aliyun-sls/opentelemetry-go-provider-sls/provider" "go.opentelemetry.io/contrib/instrumentation/net/http/otelhttp" "go.opentelemetry.io/otel" "go.opentelemetry.io/otel/label" "go.opentelemetry.io/otel/metric" "go.opentelemetry.io/otel/trace") func main() { slsConfig, err := provider.NewConfig(provider.WithServiceName("\${service}"), provider.WithServiceNamespace("\${service.namespace}"), provider.WithServiceVersion("\${version}"), provider.WithTraceExporterEndpoint("\${endpoint}"), provider.WithMetricExporterEndpoint("\${endpoint}"), provider.WithSLSConfig("\${project}", "\${instance}", "\${access-key-id}", "\${access-key-secret}")) // Invoke the panic() function. If the initialization fails, the OpenTelemetry provider exits. You can also use other error handling m ethods. if err != nil { panic(err) if err := provider.Start(slsConfig); err != nil { panic(err) defer provider.Shutdown(slsConfig) $\ensuremath{{\prime}}\xspace$ // If you want to analyze metric data in the application, you can register the metrics. labels := []label.KeyValue{ label.String("label1", "value1"), meter := otel.Meter("aliyun.sls") sayDavidCount := metric.Must(meter).NewInt64Counter("say david count") helloHandler := func(w http.ResponseWriter, req *http.Request) { if time.Now().Unix()%10 == 0 { _, _ = io.WriteString(w, "Hello, world!\n") } else { // If you want to record some events, you can obtain the span in the context and add events. ctx := req.Context() span := trace.SpanFromContext(ctx) span.AddEvent("say : Hello, I am david", trace.WithAttributes(label.KeyValue{ Key: "label-key-1", Value: label.StringValue("label-value-1"), , _ = io.WriteString(w, "Hello, I am david!\n") sayDavidCount.Add(req.Context(), 1, labels...) } } // To use the automatic instrumentation solution for otel net/http, you need to only enclose http.Handler with otelhttp.NewHandler. otelHandler := otelhttp.NewHandler(http.HandlerFunc(helloHandler), "Hello") http.Handle("/hello", otelHandler) fmt.Println("Now listen port 8080, you can visit 127.0.0.1:8080/hello .") err = http.ListenAndServe(":8080", nil) if err != nil { panic(err) } }

```
· Use the Gorilla Mux framework to import data
    The following sample code is created based on go.opentelemetry.io/contrib/instrumentation/github.com/gorilla/mux/otelmux v0.37.0 . The interface may change in later versions. For more information about the latest sample code, see otel-mux-example.
    Replace the variables in the following code with the actual values. For more information about the variables, see Variables.
     package main
      import (
          "context"
          "fmt"
          "net/http"
          "github.com/aliyun-sls/opentelemetry-go-provider-sls/provider"
          "github.com/gorilla/mux"
          "go.opentelemetry.io/contrib/instrumentation/github.com/gorilla/mux/otelmux"
          "go.opentelemetry.io/otel"
          "go.opentelemetry.io/otel/label"
          "go.opentelemetry.io/otel/metric"
          "go.opentelemetry.io/otel/trace"
      )
      func main() {
          slsConfig, err := provider.NewConfig(provider.WithServiceName("${service}"),
              provider.WithServiceNamespace("${service.namespace}"),
              provider.WithServiceVersion("${version}"),
              provider.WithTraceExporterEndpoint("${endpoint}"),
              provider.WithMetricExporterEndpoint("${endpoint}"),
          provider.WithSLSConfig("${project]", "${instance}", "${access-key-id}", "${access-key-secret}"))
// Invoke the panic() function. If the initialization fails, the OpenTelemetry provider exits. You can also use other error handling m
      ethods.
          if err != nil {
              panic(err)
          if err := provider.Start(slsConfig); err != nil {
             panic(err)
          defer provider.Shutdown(slsConfig)
          // If you want to analyze metric data in the application, you can register the metrics.
          labels := []label.KeyValue{
              label.String("label1", "value1"),
          meter := otel.Meter("aliyun.sls")
          callUsersCount := metric.Must(meter).NewInt64Counter("call users count")
          r := mux.NewRouter()
          r.Use(otelmux.Middleware("my-server"))
          r.HandleFunc("/users/{id:[0-9]+}", http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
              vars := mux.Vars(r)
              id := vars["id"]
              callUsersCount.Add(r.Context(), 1, labels...)
              name := getUser(r.Context(), id)
              reply := fmt.Sprintf("user %s (id %s)\n", name, id)
          _, _ = w.Write(([]byte)(reply))
}))
          http.Handle("/", r)
          fmt.Println("Now listen port 8080, you can visit 127.0.0.1:8080/users/xxx .")
          _ = http.ListenAndServe(":8080", nil)
      func getUser(ctx context.Context, id string) string {
          if id == "123" {
              return "otelmux tester"
          \ensuremath{{\prime}}\xspace // If you want to record some events, you can obtain the span in the context and add events.
          span := trace.SpanFromContext(ctx)
          span.AddEvent("unknown user id : "+id, trace.WithAttributes(label.KeyValue{
              Key: "label-key-1",
              Value: label.StringValue("label-value-1"),
          }))
          return "unknown"

    Manual mode

  Replace the variables in the following code with the actual values. For more information about the variables, see Variables.
```

```
// Copyright The AliyunSLS Authors
//
// Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
//
// http://www.apache.org/licenses/LICENSE-2.0
//
```

// Unless required by applicable law or agreed to in writing, software // distributed under the License is distributed on an "AS IS" BASIS, // WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. // See the License for the specific language governing permissions and // limitations under the License. package main import ("context" "errors" "fmt" "go.opentelemetry.io/otel/attribute" "go.opentelemetry.io/otel/metric/global" "go.opentelemetry.io/otel/metric/instrument/asyncfloat64" "math/rand" "time" "github.com/aliyun-sls/opentelemetry-go-provider-sls/provider" "go.opentelemetry.io/otel" "go.opentelemetry.io/otel/codes" "go.opentelemetry.io/otel/trace") func main() { slsConfig, err := provider.NewConfig(provider.WithServiceName("payment"), provider.WithServiceVersion("v0.1.0"), provider.WithTraceExporterEndpoint("stdout"), provider.WithMetricExporterEndpoint("stdout"), provider.WithSLSConfig("test-project", "test-otel", "access-key-id", "access-key-secret")) // Invoke the panic() function. If the initialization fails, the OpenTelemetry provider exits. You can also use other error handling metho ds. if err != nil { panic(err) if err := provider.Start(slsConfig); err != nil { panic(err) defer provider.Shutdown(slsConfig) mockTrace() mockMetrics() } func mockMetrics() { $\ensuremath{{//}}\xspace$ Add the label information. labels := []attribute.KeyValue{ attribute.String("label1", "value1"), meter := global.Meter("ex.com/basic") c, := meter.AsyncFloat64().Counter("randval") // The observed value, which is used to obtain a measured value on a regular basis. The callback function is invoked once per reporting cy cle. go mockObserveMetric(c, labels) temperature, _ := meter.SyncFloat64().Counter("temperature")
interrupts, _ := meter.SyncInt64().Counter("interrupts") ctx := context.Background() for { temperature.Add(ctx, 100+10*rand.NormFloat64(), labels...) interrupts.Add(ctx, int64(rand.Intn(100)), labels...) time.Sleep(time.Second * time.Duration(rand.Intn(10))) } } func mockObserveMetric(c asyncfloat64.Counter, labels []attribute.KeyValue) { timer := time.NewTimer(1 * time.Second) select { case <-timer.C: c.Observe(context.Background(), rand.Float64(), labels...) timer.Stop() } func mockTrace() { tracer := otel.Tracer("ex.com/basic") ctx0 := context.Background()

```
ctx1, finish1 := tracer.Start(ctx0, "foo")
  defer finish1.End()
  ctx2, finish2 := tracer.Start(ctx1, "bar")
  defer finish2.End()
  ctx3, finish3 := tracer.Start(ctx2, "baz")
  defer finish3.End()
 ctx := ctx3
  getSpan(ctx)
 addAttribute(ctx)
  addEvent(ctx)
  recordException(ctx)
  createChild(ctx, tracer)
}
// example of getting the current span
// Obtain the current span.
func getSpan(ctx context.Context) {
 span := trace.SpanFromContext(ctx)
 fmt.Printf("current span: %v\n", span)
}
// example of adding an attribute to a span
// Add an attribute value to the span.
func addAttribute(ctx context.Context)
 span := trace.SpanFromContext(ctx)
  span.SetAttributes(attribute.KeyValue{
    Key: "label-key-1"
    Value: attribute.StringValue("label-value-1")})
}
// example of adding an event to a span
// Add an event to the span
func addEvent(ctx context.Context) {
 span := trace.SpanFromContext(ctx)
  span.AddEvent("event1", trace.WithAttributes(
    attribute.String("event-attr1", "event-string1"),
    attribute.Int64("event-attr2", 10)))
}
// example of recording an exception
// Record the result of the span and the error information.
func recordException(ctx context.Context) {
 span := trace.SpanFromContext(ctx)
 span.RecordError(errors.New("exception has occurred"))
 span.SetStatus(codes.Error, "internal error")
// example of creating a child span % \left( {{\left( {{{\left( {{{\left( {{{\left( {{{c}}} \right)}} \right.}} \right)}_{i}}} \right)}_{i}} \right)} \right)
// Create a child span.
func createChild(ctx context.Context, tracer trace.Tracer) {
 // span := trace.SpanFromContext(ctx)
  _, childSpan := tracer.Start(ctx, "child")
  defer childSpan.End()
  fmt.Printf("child span: %v\n", childSpan)
}
```

What to do next

```
• View the details of a trace instance
```

• Query and analyze trace data

4.9.1.4.2.3. Import trace data from Python applications to Log Service by using

OpenTelemetry SDK for Python

This topic describes how to import trace data from Python applications to Log Service by using OpenTelemetry SDK for Python.

Prerequisites

- A trace instance is created. For more information, see Create a trace instance.
- A Python development environment is set up. The Python version is 3.7 or later.
- OpenTelemetry SDK for Python is installed in your Python development environment. If OpenTelemetry SDK for Python is not installed, you can run the following commands to install the SDK: pip install opentelemetry-api==1.12.0

```
pip install opentelemetry-sdk==1.12.0
pip install opentelemetry-exporter-otlp==1.12.0
```

Procedure

- 1. Initialize an OpenTelemetry provider.
- 2. Check whether the conditions for importing data in semi-automatic mode are met.
 - If the conditions are met you can import trace data in semi-automatic mode

In the conditions are met, you can import trace data in semi-datomatic mode.

- For scenarios in which trace data cannot be imported in semi-automatic mode, you must import the data in manual mode.
- $\circ~$ If the conditions are not met, you can import trace data in manual mode.

Step 1: Initialize an OpenTelemetry provider

You can initialize an OpenTelemetry provider by using the following code. You must replace the variables in the code with the actual values. For more information about the variables, see Variables.

```
# For Opentelemetry
import socket
from opentelemetry import trace
from \ opentelemetry.exporter.otlp.proto.grpc.trace\_exporter \ import \ OTLPSpanExporter
from opentelemetry.sdk.resources import Resource
from opentelemetry.sdk.trace import TracerProvider
from opentelemetry.sdk.trace.export import BatchSpanProcessor
from opentelemetry.sdk.trace.export import ConsoleSpanExporter
from opentelemetry.sdk.trace.export import SimpleSpanProcessor
class OpenTelemetrySLSProvider(object):
    def __init__(self, namespace="", service="", version="", endpoint='stdout',
                project=None, instance=None, access_key_id=None, access_key_secret=None):
        :param namespace: Your service namespace
        :param service: Your Application Service Nam
       :param version: Your Application Version
        :param endpoint: console or https://sls endpoint:10010
       :param project: SLS project
        :param instance: SLS OTEL InstanceId
        :param access_key_id: Aliyun AccesskeyId
        :param access_key_secret: Aliyun AccesskeySecret
       self.sls_otel_endpoint = endpoint
        self.sls_otel_project = project
        self.sls_otel_akid = access_key_id
        self.sls_otel_aksecret = access_key_secret
        self.sls otel instanceid = instance
        self.local mode = False
        if endpoint == "stdout":
            self.local_mode = True
            self.resource = Resource(attributes={
                "host.name": socket.gethostname(),
                "service.name": service,
                "service.namespace": namespace,
                "service.version": version})
       else:
            self.resource = Resource(attributes={
                "host.name": socket.gethostname(),
                "service.namespace": namespace,
                "service.name": service,
                "service.version": version
                "sls.otel.project": self.sls_otel_project,
                "sls.otel.akid": self.sls_otel_akid,
                "sls.otel.aksecret": self.sls_otel_aksecret,
                "sls.otel.instanceid": self.sls_otel_instanceid
            })
   def initTracer(self):
        trace.set_tracer_provider(TracerProvider(resource=self.resource))
        if self.local_mode:
            trace.get_tracer_provider().add_span_processor(SimpleSpanProcessor(ConsoleSpanExporter()))
        else:
           otlp exporter = OTLPSpanExporter(endpoint=self.sls otel endpoint)
           trace.get_tracer_provider().add_span_processor(BatchSpanProcessor(otlp_exporter))
# debug mode
#sls_ot_provider = OpenTelemetrySLSProvider(service="example", version="v0.1.0")
# write to sls
sls_ot_provider = OpenTelemetrySLSProvider(namespace="${service.namespace}", service="${service}", version="${version}",
                                         endpoint='${endpoint}',
                                         project="${project}",
                                         instance="${instance}",
                                         access_key_id="${access-key-id}",
                                         access_key_secret="${access-key-secret}"
```

Table 1. Variables

Variable	Description	Example
\${service.namespace}	The namespace to which the service belongs.	order
<i>\${service}</i>	The name of the service. Specify the value based on your business requirements.	payment

<i>\${version}</i>	The version of the service. We recommend that you specify a version in the $\ensuremath{\textit{va.b.c}}$ format.	v0.1.2
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. The fixed port number is 10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations.	
	 Note If you set the variable tostdout (endpoint='stdout'), data is displayed to the standard output. If you leave the variable empty, trace data is not uploaded to Log Service. 	https://trace.test-region-id.sls-pub.test- inter-domain:10010
<i>\${project}</i>	The name of the Log Service project.	test-project
<i>\${instance}</i>	The ID of the trace instance. For more information, seeCreate a trace instance.	test-traces
\${access-key-id}	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 	None
\${access-key-secret}	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None

Step 2: Import data

You can import trace data to Log Service in semi-automatic or manual mode. OpenTelemetry SDK for Python provides various types of instrumentation packages and supports the semi-automatic instrumentation of common frameworks. If you use one of the instrumentation packages, you must import data in semi-automatic mode. For more information, visit GitHub.

Semi-automatic instrumentation

In this example, the flask and requests instrumentation packages are used.

i. Install the instrumentation packages.

```
pip install requests
pip install flask
pip install opentelemetry-instrumentation-flask
pip install opentelemetry-instrumentation-requests
```

ii. Run code.

Replace the variables in the following code with the actual values. For more information, see Variables.

```
# for flask
import flask
import requests
# for Opentelemetry instrumentation
import socket
from opentelemetry.instrumentation.flask import FlaskInstrumentor
from opentelemetry.instrumentation.requests import RequestsInstrumentor
# For Opentelemetry pip install opentelemetry-instrumentation-requests
from opentelemetry import trace
from opentelemetry.exporter.otlp.proto.grpc.trace_exporter import (
   OTLPSpanExporter,
from opentelemetry.sdk.resources import Resource
from opentelemetry.sdk.trace import TracerProvider
from opentelemetry.sdk.trace.export import BatchSpanProcessor
from opentelemetry.sdk.trace.export import ConsoleSpanExporter
from opentelemetry.sdk.trace.export import SimpleSpanPro
class OpenTelemetrySLSProvider(object):
    def __init__(self, namespace="", service="", version="", endpoint='stdout',
                project=None, instance=None, access_key_id=None, access_key_secret=None):
       :param namespace: Your service namespace
        :param service: Your Application Service Name
        :param version: Your Application Version
        :param endpoint: console or https://sls_endpoint:10010
       :param project: SLS project
        :param instance: SLS OTEL InstanceId
       :param access_key_id: Aliyun AccesskeyId
        :param access_key_secret: Aliyun AccesskeySecret
       self.sls_otel_endpoint = endpoint
       self.sls_otel_project = project
        self.sls_otel_akid = access_key_id
        self.sls_otel_aksecret = access_key_secret
        self.sls otel instanceid = instance
```

self.local_mode = False if endpoint == "stdout": self.local_mode = True self.resource = Resource(attributes={ "host.name": socket.gethostname(), "service.name": service, "service.namespace": namespace, "service.version": version}) else: self.resource = Resource(attributes={ "host.name": socket.gethostname(), "service.name": service, "service.version": version, "service.namespace": namespace, "sls.otel.project": self.sls_otel_project, "sls.otel.akid": self.sls_otel_akid, "sls.otel.aksecret": self.sls_otel_aksecret, "sls.otel.instanceid": self.sls_otel_instanceid }) def initTracer(self): trace.set_tracer_provider(TracerProvider(resource=self.resource)) if self.local_mod trace.get_tracer_provider().add_span_processor(SimpleSpanProcessor(ConsoleSpanExporter())) else: otlp_exporter = OTLPSpanExporter(endpoint=self.sls_otel_endpoint) trace.get_tracer_provider().add_span_processor(BatchSpanProcessor(otlp_exporter)) # write to sls sls_ot_provider = OpenTelemetrySLSProvider(namespace="\${service.namespace}", service="\${service}", version="\${version}", endpoint='\${endpoint}', project="\${project}", instance="\${instance}", access_key_id="\${access-key-id}", access_key_secret="\${access-key-secret}" # for console debug #sls_ot_provider = OpenTelemetrySLSProvider(service="example", version="v0.1.0") sls_ot_provider.initTracer() # flask init app = flask.Flask(__name__) # instrumentation init FlaskInstrumentor().instrument_app(app) RequestsInstrumentor().instrument() @app.route("/") def hello(): tracer = trace.get_tracer(__name__) with tracer.start_as_current_span("request_server"): requests.get("http://www.taobao.com") return "hello" app.run(debug=True, port=5000) iii. Access the service to trigger trace data generation and send the trace data to Log Service. 127.0.0.1:5000/hello

Manual instrumentation

Run the following code. Replace the variables in the code with the actual values. For more information about the code, see Variables.

ot-manual-example.py import time # For Opentelemetry import socket from opentelemetry import trace from pentelemetry.exporter.otlp.proto.grpc.trace_exporter import OTLPSpanExporter from opentelemetry.sdk.resources import Resource from opentelemetry.sdk.trace import TracerProvider from opentelemetry.sdk.trace.export import BatchSpanProcessor from opentelemetry.sdk.trace.export import ConsoleSpanExporter from opentelemetry.sdk.trace.export import SimpleSpanProcessor class OpenTelemetrySLSProvider(object): def __init__(self, namespace="", service="", version="", endpoint='stdout', project=None, instance=None, access_key_id=None, access_key_secret=None): ... :param service: Your service namespace :param service: Your Application Service Name :param version: Your Application Version :param endpoint: console or https://sls endpoint:10010 :param project: SLS project :param instance: SLS OTEL InstanceId :param access_key_id: Aliyun AccesskeyId :param access_key_secret: Aliyun AccesskeySecret self.sls_otel_endpoint = endpoint self.sls_otel_project = project self.sls_otel_akid = access_key_id self.sls_otel_aksecret = access_key_secret self.sls_otel instanceid = instance self.local_mode = False if endpoint == "stdout": self.local_mode = True self.resource = Resource(attributes={ "host.name": socket.gethostname(), "service.name": service, "service.namespace": namespace, "service.version": version}) else: self.resource = Resource(attributes={ "host.name": socket.gethostname(), "service.name": service, "service.version": version, "service.namespace": namespace, "sls.otel.project": self.sls_otel_project, "sls.otel.akid": self.sls_otel_akid, "sls.otel.aksecret": self.sls_otel_aksecret, "sls.otel.instanceid": self.sls_otel_instanceid def initTracer(self): trace.set_tracer_provider(TracerProvider(resource=self.resource)) if self.local_mode: trace.get_tracer_provider().add_span_processor(SimpleSpanProcessor(ConsoleSpanExporter())) else: otlp exporter = OTLPSpanExporter(endpoint=self.sls otel endpoint) trace.get_tracer_provider().add_span_processor(BatchSpanProcessor(otlp_exporter)) # write to sls sls_ot_provider = OpenTelemetrySLSProvider(namespace="\${service.namespace}", service="\${service}", version="\${version}", endpoint='\${endpoint}', project="\${project}", instance="\${instance}", access_key_id="\${access-key-id}", access_key_secret="\${access-key-secret}" # for console debug #sls_ot_provider = OpenTelemetrySLSProvider(service="example", version="v0.1.0") # Trace Example sls_ot_provider.initTracer() tracer = trace.get_tracer(__name__) with tracer.start_as_current_span("foo"): print("Hello world!") labels = {"environment": "staging"} requests_counter.add(25, labels) time.sleep(60)

FAQ

How do I check whether OpenTelemetry SDK for Python is installed as expected? You can check whether the SDK is installed as expected by using the following sample code. Save the code as a tracing.py file. Then, run the tracing.py command. If the command output is returned as expected, the related dependencies of OpenTelemetry SDK for Python are installed. # tracing-example-1.py from opentelemetry import trace from opentelemetry.sdk.trace import TracerProvider from opentelemetry.sdk.trace.export import (ConsoleSpanExporter, SimpleSpanProcessor,) trace.set_tracer_provider(TracerProvider()) trace.get_tracer_provider().add_span_processor(SimpleSpanProcessor (ConsoleSpanExporter())) tracer = trace.get tracer(name) with tracer.start_as_current_span("foo"): print("Hello world!") Hello world! ł "name": "foo", "context": { "trace_id": "0x52 706e3eeec9f2b79158", "span_id": "0xa4 "trace_state": "[]" "kind": "SpanKind.INTERNAL", "parent_id": null, "start_time": "2021-02-24T03:58:36.377024Z", "end_time": "2021-02-24T03:58:36.377133Z", "status": { "status_code": "UNSET" }, "attributes": {}, "events": [], "links": [], "resource": { "telemetry.sdk.language": "python", "telemetry.sdk.name": "opentelemetry", "telemetry.sdk.version": "0.17b0" }

What to do next

View the details of a trace instance

Query and analyze trace data

4.9.1.4.2.4. Import trace data from Node.js applications to Log Service by using

OpenTelemetry SDK for JavaScript

This topic describes how to import trace data from Node.js applications to Log Service by using OpenTelemetry SDK for JavaScript.

Prerequisites

- A trace instance is created. For more information, see Create a trace instance.
- A Node.js development environment is set up. The Node.js version is 8.5.0 or later.

Method 1: (Recommended) Import data in semi-automatic mode

Some frameworks support automatically uploading trace data to Log Service. You can implement this function by installing node.js dependencies on these frameworks, which include HTTP, HTTPS, gRPC, Express, MySQL, MongoDB, and Redis. For more information about the frameworks, see opentelemetry-node-js-contrib. In this example, Express is used to describe how to import data in semi-automatic mode. For more information, see Examples.

1. Install dependencies.

npm install --save @opentelemetry/api npm install --save @opentelemetry/node npm install --save @opentelemetry/tracing npm install --save @opentelemetry/instrumentation npm install --save @opentelemetry/instrumentation-express npm install --save @opentelemetry/instrumentation-http npm install --save @grpc/grpc-js npm install --save @opentelemetry/sdk-trace-node

2. Initialize a tracer and start Express.

Replace the variables in the following code with the actual values. For more information about the variables, see Variables.

User Guide-Log Service

const opentelemetry = require("@opentelemetry/api"); const { registerInstrumentations } = require("@opentelemetry/instrumentation"); const { NodeTracerProvider } = require("@opentelemetry/sdk-trace-node"); const { Resource } = require("@opentelemetry/resources"); const { SemanticResourceAttributes, } = require("@opentelemetry/semantic-conventions"); const { SimpleSpanProcessor, ConsoleSpanExporter, } = require("@opentelemetry/tracing"); const grpc = require("@grpc/grpc-js"); const { CollectorTraceExporter, } = require("@opentelemetry/exporter-collector-grpc"); const { ExpressInstrumentation, } = require("@opentelemetry/instrumentation-express"); const { HttpInstrumentation } = require("@opentelemetry/instrumentation-http"); var os = require("os"); var hostname = os.hostname(); const provider = new NodeTracerProvider({ resource: new Resource({ [SemanticResourceAttributes.SERVICE NAME]: "\${service}", [SemanticResourceAttributes.DEPLOYMENT_ENVIRONMENT]: "\${environment}", [SemanticResourceAttributes.SERVICE_VERSION]: "\${version}", [SemanticResourceAttributes.SERVICE_NAMESPACE]: "\${service.namespace}", [SemanticResourceAttributes.HOST_NAME]: hostname, }), }); provider.register(); registerInstrumentations({ instrumentations: [new HttpInstrumentation(), new ExpressInstrumentation({ ignoreLayersType: [new RegExp("middleware.*")], }),], tracerProvider: provider, }); var url = "\${endpoint}"; var logStdout = false; if (url == "stdout") { logStdout = true; var meta = new grpc.Metadata(); meta.add("x-sls-otel-project", "\${project}");
meta.add("x-sls-otel-instance-id", "\${instance}"); meta.add("x-sls-otel-ak-id", "\${access-key-id}"); meta.add("x-sls-otel-ak-secret", "\${access-key-secret}"); const collectorOptions = { url: url, credentials: grpc.credentials.createSsl(), metadata: meta, }; const exporter = new CollectorTraceExporter(collectorOptions); if (!logStdout) { provider.addSpanProcessor(new SimpleSpanProcessor(exporter)); } else { var stdexporter = new ConsoleSpanExporter(); provider.addSpanProcessor(new SimpleSpanProcessor(stdexporter)); } provider.register(); var tracer = opentelemetry.trace.getTracer("\${service}"); var express = require("express"); var app = express(); app.get("/hello", function (req, res, next) { res.send("success"); }); var server = app.listen(8079, function () { var port = server.address().port; console.log("App now running in %s mode on port %d", app.get("env"), port); });

Table 1. Variables

Variable	Description	Example
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. The fixed port number is 10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations.	trace.test-region-id.sls-pub.test-inter- domain:10010
<i>\${project}</i>	The name of the Log Service project.	test-project
<i>\${instance}</i>	The ID of the trace instance. For more information, seeCreate a trace instance.	test-traces
\${access-key-id}	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. o For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 	None
\${access-key-secret}	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None
<i>\${service}</i>	The name of the service. Specify the value based on your business requirements.	payment
\${version}	The version of the service. We recommend that you specify a version in the $\ensuremath{\mathbf{va.b.c}}$ format.	v0.1.2
\${service.namespace}	The namespace to which the service belongs.	order
\${environment}	The deployment environment of the service. Examples: test environment, staging environment, or production environment.	pre

3. Access the service to trigger trace data generation and send the trace data to Log Service.

127.0.0.1:8079/hello

Method 2: Manually generate and send trace data

If you use a self-managed framework or have special requirements, you can manually construct trace data and send the data to Log Service. For more information, see opentelemetry-js.

1. Install dependencies.

```
npm install --save @opentelemetry/api
npm install --save @opentelemetry/node
npm install --save @opentelemetry/tracing
npm install --save @opentelemetry/exporter-collector-grpc
```

2. Initialize a tracer and start Express.

Replace the variables in the following code with the actual values. For more information, see Variables.

```
const opentelemetry = require("@opentelemetry/api");
const { registerInstrumentations } = require("@opentelemetry/instrumentation");
const { NodeTracerProvider } = require("@opentelemetry/sdk-trace-node");
const { Resource } = require("@opentelemetry/resources");
const {
 SemanticResourceAttributes,
} = require("@opentelemetry/semantic-conventions");
const {
SimpleSpanProcessor,
 ConsoleSpanExporter,
} = require("@opentelemetry/tracing");
const grpc = require("@grpc/grpc-js");
const {
CollectorTraceExporter,
} = require("@opentelemetry/exporter-collector-grpc");
const {
ExpressInstrumentation,
} = require("@opentelemetry/instrumentation-express");
const { HttpInstrumentation } = require("@opentelemetry/instrumentation-http");
var os = require("os");
var hostname = os.hostname();
const provider = new NodeTracerProvider({
 resource: new Resource({
    [SemanticResourceAttributes.SERVICE_NAME]: "${service}",
     [SemanticResourceAttributes.DEPLOYMENT_ENVIRONMENT]: "${environment}",
   [SemanticResourceAttributes.SERVICE_VERSION]: "${version}",
    [SemanticResourceAttributes.SERVICE_NAMESPACE]: "${service.namespace}",
    [SemanticResourceAttributes.HOST_NAME]: hostname,
 }),
});
provider.register();
registerInstrumentations({
instrumentations: [
```

```
new HttpInstrumentation(),
        new ExpressInstrumentation({
         ignoreLayersType: [new RegExp("middleware.*")],
       }),
     1,
     tracerProvider: provider,
    });
    var url = "${endpoint}";
    var logStdout = false;
    if (url == "stdout") {
     logStdout = true;
    var meta = new grpc.Metadata();
    meta.add("x-sls-otel-project", "${project}");
    meta.add("x-sls-otel-instance-id", "${instance}");
    meta.add("x-sls-otel-ak-id", "${access-key-id}");
    meta.add("x-sls-otel-ak-secret", "${access-key-secret}");
    const collectorOptions = {
     url: url,
     credentials: grpc.credentials.createSsl(),
     metadata: meta,
    };
    const exporter = new CollectorTraceExporter(collectorOptions);
    if (!logStdout) {
     provider.addSpanProcessor(new SimpleSpanProcessor(exporter));
    } else {
     var stdexporter = new ConsoleSpanExporter();
     provider.addSpanProcessor(new SimpleSpanProcessor(stdexporter));
    }
    provider.register();
    var tracer = opentelemetry.trace.getTracer("${service}");
    var express = require('express');
    var app = express()
    app.get('/hello', function (req, res, next) {
        const span = tracer.startSpan('hello');
        span.setAttribute('name', 'toma');
span.setAttribute('age', '26');
        span.addEvent('invoking doWork');
       res.send("success");
        span.end();
    });
    var server = app.listen(8079, function () {
      var port = server.address().port;
      console.log("App now running in %s mode on port %d", app.get("env"), port);
    });
3. Access the service to trigger trace data generation and send the trace data to Log Service.
   127.0.0.1:8079/hello
```

What to do next

• View the details of a trace instance

• Query and analyze trace data

4.9.1.4.2.5. Import trace data from C# applications to Log Service by using

OpenTelemetry SDK for .NET

This topic describes how to import trace data from C# applications to Log Service by using OpenTelemetry SDK for .NET.

Prerequisites

- A trace instance is created. For more information, see Create a trace instance.
- A .NET Framework development environment is set up

⑦ Note The import method in this topic supports all official versions of .NET Framework except for .NET Framework 3.5 SP1. For more information, see .NET Core and .NET Framework.

Procedure

1. Add dependencies.

```
dotnet add package OpenTelemetry --version 1.2.0-beta1
    dotnet add package OpenTelemetry.Exporter.Console --version 1.2.0-beta1
    dotnet add package OpenTelemetry.Exporter.OpenTelemetryProtocol --version 1.2.0-beta1
    dotnet add package OpenTelemetry.Extensions.Hosting --version 1.0.0-rc8
    dotnet add package OpenTelemetry.Instrumentation.AspNetCore --version 1.0.0-rc8
    dotnet add package Grpc.Core --version 2.36.4
2. Run code.
   Replace the variables in the following code with the actual values. For more information about the variables, see Variables.
    using System;
    using OpenTelemetry;
    using OpenTelemetry.Trace;
    using System.Diagnostics;
    using System.Collections.Generic;
    using OpenTelemetry.Resources;
    using Grpc.Core;
    namespace mydemo
        class Program
        {
              private static readonly ActivitySource MyActivitySource = new ActivitySource(
            "MyCompany.MyProduct.MyLibrary");
            static void Main(string[] args) {
                using var tracerProvider = Sdk.CreateTracerProviderBuilder()
                .SetSampler(new AlwaysOnSampler())
                .AddSource("MyCompany.MyProduct.MyLibrary")
                .AddOtlpExporter(opt => {
                    opt.Endpoint = new Uri("${endpoint}");
opt.Headers = "x-sls-otel-project=${project},x-sls-otel-instance-id=${instance},x-sls-otel-ak-id=${access-key-id},x-sls-otel-
    ak-secret=${access-key-secret}";
               })
                 .SetResourceBuilder(OpenTelemetry.Resources.ResourceBuilder.CreateDefault()
                .AddAttributes(new Dictionary<string, object> { { { "service.name", "${service}" },
                     {"service.version","${version}"},
                     {"service.host","${host}"},
                    {"service.namespace","${service.namespace}"}
                    }))
                .Build();
                 using (var activity = MyActivitySource.StartActivity("SayHello"))
                     activity?.SetTag("foo", 1);
activity?.SetTag("bar", "Hello, World!");
                     activity?.SetTag("baz", new int[] { 1, 2, 3 });
                 Console.WriteLine("Hello World!");
            }
        }
```

Table 1. Variables

}

Variable	Description	Example	
\${service}	The name of the service. Specify the value based on your business requirements.	payment	
\${version}	The version of the service. We recommend that you specify a version in the $\ensuremath{\textbf{va.b.c}}$ format.	v0.1.2	
\${host}	The hostname.	localhost	
\${endpoint}	The endpoint of the Log Service project. Format: https://trace.\${region_id}.sls-pub.\${internet-domain}:10010. The fixed port number is 10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations.	https://trace.test-region-id.sls- pub.test-inter-domain:10010	
\${project}	The name of the Log Service project.	test-project	
<i>\${instance}</i>	The ID of the trace instance. For more information, seeCreate a trace instance.	instance-traces	
\${access-key-id}	The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. • For information about how to grant a RAM user the write permissions on the	LTAI4Fvyv****	
	 Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 		
\${access-key-secret}	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	HfJEw25sYldO***	

order

\${service.namespace}

The namespace to which the service belongs.

What to do next

- View the details of a trace instance
- Query and analyze trace data

4.9.1.4.2.6. Import trace data from Rust applications to Log Service by using

OpenTelemetry SDK for Rust

This topic describes how to import trace data from Rust applications to Log Service by using OpenTelemetry SDK for Rust.

Prerequisites

- A trace instance is created. For more information, see Create a trace instance.
- A Rust development environment is set up. The Rust version is 1.46 or later.

Procedure

1. Add dependencies.

```
[package]
name = "test"
version = "0.1.0"
authors = [""]
edition = "2018"

# See more keys and their definitions at The Manifest Format.
[dependencies]
futures = "0.3"
lazy_static = "1.4"
opentelemetry = { version = "0.16.0", features = ["tokio-support", "metrics", "serialize"] }
opentelemetry = { version = "0.9.0", features = ["tonic", "metrics", "tls", "tls-roots"] }
serde_json = "1.0"
tokio = { version = "1.0", features = ["full"] }
tonic="0.4.0"
url = "2.2.0"
```

2. Run code.

Replace the variables in the following code with the actual values. For more information about the variables, see Variables.

```
use opentelemetry::global::shutdown_tracer_provider;
use opentelemetry::sdk::Resource;
use opentelemetry::trace::TraceError;
use opentelemetry::{
   baggage::BaggageExt,
    trace::{TraceContextExt, Tracer},
   Context, Key, KeyValue,
};
use opentelemetry::{global, sdk::trace as sdktrace};
use opentelemetry_otlp::WithExportConfig;
use std::error::Error;
use std::time::Duration;
use tonic::metadata::MetadataMap;
use tonic::transport::ClientTlsConfig;
use url::Url;
static ENDPOINT: &str = "https://${endpoint}";
static PROJECT: &str = "${project}";
static INSTANCE_ID: &str = "${instance}";
static AK_ID: &str = "${access-key-id}";
static AK_SECRET: &str = "${access-key-secret}";
static SERVICE_VERSION: &str = "${version}";
static SERVICE NAME: &str = "${service}";
static SERVICE NAMESPACE: &str = "${service.namespace}";
static HOST_NAME: &str = "${host}";
static SLS_PROJECT_HEADER: &str = "x-sls-otel-project";
static SLS INSTANCE ID HEADER: &str = "x-sls-otel-instance-id";
static SLS AK ID HEADER: &str = "x-sls-otel-ak-id";
static SLS_AK_SECRET_HEADER: &str = "x-sls-otel-ak-secret";
static SLS_SERVICE_VERSION: &str = "service.version";
static SLS_SERVICE_NAME: &str = "service.name";
static SLS SERVICE NAMESPACE: &str = "service.namespace";
static SLS HOST NAME: &str = "host.name";
fn init_tracer() -> Result<sdktrace::Tracer, TraceError> {
   let mut metadata_map = MetadataMap::with_capacity(4);
    metadata_map.insert(SLS_PROJECT_HEADER, PROJECT.parse().unwrap());
   metadata_map.insert(SLS_INSTANCE_ID_HEADER, INSTANCE_ID.parse().unwrap());
    metadata map.insert(SLS AK ID HEADER, AK ID.parse().unwrap());
   metadata_map.insert(SLS_AK_SECRET_HEADER, AK_SECRET.parse().unwrap());
    let endpoint = ENDPOINT;
    let endpoint = Url::parse(&endpoint).expect("endpoint is not a valid url");
    let resource = vec![
```

```
KeyValue::new(SLS_SERVICE_VERSION, SERVICE_VERSION),
         KeyValue::new(SLS HOST NAME, HOST NAME),
         KeyValue::new(SLS SERVICE NAMESPACE, SERVICE NAMESPACE),
         KeyValue::new(SLS_SERVICE_NAME, SERVICE_NAME),
     1;
     opentelemetry_otlp::new_pipeline()
         .tracing()
         .with_exporter(
             opentelemetry_otlp::new_exporter()
                .tonic()
                 .with endpoint(endpoint.as str())
                 .with_metadata(dbg!(metadata_map))
                 .with_tls_config(
                     ClientTlsConfig::new().domain_name(
                         endpoint
                             .host str()
                             .expect("the specified endpoint should have a valid host"),
                    ),
                 ),
         )
         .with trace config(sdktrace::config().with resource(Resource::new(resource)))
         .install batch (opentelemetry::runtime::Tokio)
 }
 const FOO_KEY: Key = Key::from_static_str("ex.com/foo");
 const BAR_KEY: Key = Key::from_static_str("ex.com/bar");
const LEMONS_KEY: Key = Key::from_static_str("lemons");
const ANOTHER_KEY: Key = Key::from_static_str("ex.com/another");
 lazy_static::lazy_static! {
     static ref COMMON_ATTRIBUTES: [KeyValue; 4] = [
        LEMONS KEY.164(10),
         KeyValue::new("A", "1"),
         KeyValue::new("B", "2"),
         KeyValue::new("C", "3"),
    ];
 }
 #[tokio::main]
 async fn main() -> Result<(), Box<dyn Error + Send + Sync + 'static>> {
    let _ = init_tracer()?;
     let tracer = global::tracer("ex.com/basic");
     let baggage =
         Context::current_with_baggage(vec![FOO_KEY.string("foo1"), BAR_KEY.string("bar1")])
             .attach();
     tracer.in span("operation", |cx| {
        let span = cx.span();
         span.add_event(
             "Nice operation!".to_string(),
             vec![Key::new("bogons").i64(100)],
         );
         span.set_attribute(ANOTHER_KEY.string("yes"));
         tracer.in_span("Sub operation...", |cx| {
             let span = cx.span();
             span.set_attribute(LEMONS_KEY.string("five"));
             span.add_event("Sub span event".to_string(), vec![]);
         });
    });
     tokio::time::sleep(Duration::from secs(60)).await;
     shutdown_tracer_provider();
     Ok(())
Table 1. Variables
```

Variable Description Example \${service} The name of the service. Specify the value based on your business requirements. payment The version of the service. We recommend that you specify a version in the \${version} v0.1.2 va.b.c format. The namespace to which the service belongs. \${service.namespace} order \${host} The hostname. localhost The endpoint of the Log Service project. Format:**trace.\${region_id}.sls-pub.\${internet-domain}:10010**. The endpoint of the region where the Log Service project resides. For more information, see the **Obtain the endpoint of** trace.test-region-id.sls-pub.test-internet-domain:10010 \${endpoint} Log Service topic in Preparations. \${project} The name of the Log Service project. test-project \${instance} The ID of the trace instance. For more information, seeCreate a trace instance. test-traces
\${access-key-id}	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 	None	
\${access-key-so	ecret}	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None

What to do next

- View the details of a trace instance
- Query and analyze trace data

4.9.1.4.2.7. Import trace data from Ruby applications to Log Service by using

OpenTelemetry SDK for Ruby

This topic describes how to import trace data from Ruby applications to Log Service by using OpenTelemetry SDK for Ruby.

Prerequisites

- A trace instance is created. For more information, see Create a trace instance.
- A Ruby development environment is set up. The Ruby version is 2.0 or later.

OpenTelemetry SDK for Ruby is installed.
 If OpenTelemetry SDK for Ruby is not installed, you can run the following commands to install the SDK:

gem install opentelemetry-api gem install opentelemetry-sdk gem install opentelemetry-exporter-otlp

Procedure

1. Run the following command to configure environment variables:

Replace the variables in the following code with the actual values. For more information about the variables, see Variables.

export

 $\label{eq:constraint} \texttt{OTEL_RESOURCE_ATTRIBUTES=sls.otel.project=} \\ \texttt{project}, \texttt{sls.otel.instanceid=} \\ \texttt{sls.otel.akid=} \\ \texttt{sls.otel.akid}, \texttt{sls.otel.aksecret} \\ \texttt{sls.otel.akid}, \texttt{sls.otel.aksecret} \\ \texttt{sls.otel.akid} \\ \texttt{sls.otel$

export

 $\label{eq:constraint} \texttt{OTEL}_\texttt{RESOURCE}_\texttt{ATTRIBUTES} = \texttt{sls.otel.project} \\ \texttt{sls.otel.akid} \\ \texttt{sls.o$ mespace=\${service.namespace},service.name=\${service},service.version=\${version},host.name=\${host}

Table 1. Variables

Variable	Description	Example
<i>\${service}</i>	The name of the service. Specify the value based on your business requirements.	payment
\${version}	The version of the service. We recommend that you specify a version in the $\ensuremath{\textbf{va.b.c}}$ format.	v0.1.2
\${service.namespace}	The namespace to which the service belongs.	order
<i>\${project}</i>	The name of the Log Service project.	test-project
<i>\${instance}</i>	The ID of the trace instance. For more information, seeCreate a trace instance.	test-traces
\${akid}	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 	None
\${aksecret}	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None
\${host}	The hostname.	localhost

2. Configure instrumentation.

Replace the \${endpoint} variables, see Variables. For more information about the variables, see Variables. For more information about sample code, see opentelemetry-rub

Cloud Defined Storage

require 'opentelemetry/sdk'
require 'opentelemetry-exporter-otlp'
Configure the sdk with default export and context propagation formats
see SDK#configure for customizing the setup
OpenTelemetry::SDK.configure do c
c.add_span_processor(
OpenTelemetry::SDK::Trace::Export::BatchSpanProcessor.new(
OpenTelemetry::Exporter::OTLP::Exporter.new(
endpoint: 'https://\${endpoint}/opentelemetry/v1/traces'
)
)
)
end
To start a trace you need to get a Tracer from the TracerProvider
<pre>tracer = OpenTelemetry.tracer_provider.tracer('my_app_or_gem', '0.1.0')</pre>
tracer.in_span('foo') do span
set an attribute
<pre>span.set_attribute('tform', 'osx')</pre>
add an event
span.add_event('event in bar')
create bar as child of foo
tracer.in_span('bar') do child_span
inspect the span
pp child_span
end
end

sleep 10

Table 2. Variables

Variable	Description	Example
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations.	https://trace.test-region-id.sls- pub.test-internet-domain:10010

What to do next

- View the details of a trace instance
- Query and analyze trace data

4.9.1.4.2.8. Import trace data from PHP applications to Log Service by using

Zipkin

This topic describes how to import trace data from PHP applications to Log Service by using Zipkin.

Prerequisites

- A trace instance is created. For more information, see Create a trace instance.
- PHP is installed.
- Composer is installed.

Procedure

- 1. Click here to download the official sample code of Zipkin.
- 2. Modify the parameters in the functions.php file.

 - Modify the \$httpReporterURL parameter. Replace the \${endpoint} variable in the code with the actual value. For more information about the variables, see Variables.

\$httpReporterURL = 'https://\${endpoint}/zipkin/api/v2/spans';

T - 1-1 -	1	\/
rable	Τ.	variables

Variable	Description	Example
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations.	trace.test-region-id.sls-pub.test- internet-domain:10010



Replace the variables in the following code with the actual values. For more information about the variables, see Variables.

Table 2. Variables

Variable	Description	Example
\${project}	The name of the Log Service project.	test-project
<i>\${instance}</i>	The ID of the trace instance. For more information, seeCreate a trace instance.	test-traces
\${access-key-id}	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 	None
\${access-key-secret}	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None

3. Install dependencies.

composer install

4. Start the service.

composer run-frontend composer run-backend

- 5. Access the service and then send the trace data to Log Service.
- curl http://localhost:8081

What to do next

- View the details of a trace instance
- Query and analyze trace data

4.9.1.4.2.9. Import trace data from C++ applications to Log Service by using

Jaeger SDK for C++

This topic describes how to import trace data from C++ applications to Log Service by using Jaeger SDK for C++.

Prerequisites

- A trace instance is created. For more information, see Create a trace instance.
- A development environment in which Jaeger SDK for C++ can be compiled and run is prepared.
- If you use the CMake Editor, the version must be 3.0 or later.
- $\circ~$ If you use the GCC or g++ compiler, the version must be 4.9.0 or later.

Procedure

- 1. Download and compile the SDK.
 - i. Click here to download Jaeger SDK for C++.
- ii. Decompress the package to the specified path.
- iii. Go to the specified path after the package is decompressed, and run the following commands:

```
mkdir build
cd build
CXXFLAGS="-Wno-error=deprecated-copy" cmake ..
make
```

2. Compile and run the code.

```
i. Modify the App.cpp file of the examples directory.
Replace the content of the App.cpp file with the following content. The following content indicates that Jaeger is initialized by using environment
   variables. For more information, see jaeger-client-cpp
    #include <iostream>
    #include <jaegertracing/Tracer.h>
    #include <jaegertracing/utils/EnvVariable.h>
    namespace {
    void setUpTracer()
         const auto serviceName = jaegertracing::utils::EnvVariable::getStringVariable("JAEGER_SERVICE_NAME");
         auto config = jaegertracing::Config();
        config.fromEnv();
        auto tracer = jaegertracing::Tracer::make(
            serviceName, config, jaegertracing::logging::consoleLogger());
         opentracing::Tracer::InitGlobal(
             std::static_pointer_cast<opentracing::Tracer>(tracer));
    void tracedSubroutine(const std::unique_ptr<opentracing::Span>& parentSpan)
         auto span = opentracing::Tracer::Global()->StartSpan(
    "tracedSubroutine", { opentracing::ChildOf(&parentSpan->context()) });
    void tracedFunction()
         auto span = opentracing::Tracer::Global() ->StartSpan("tracedFunction");
         tracedSubroutine(span);
    } // anonymous namespace
    int main(int argc, char* argv[])
         setUpTracer();
         tracedFunction();
        // Not stricly necessary to close tracer, but might flush any buffered
// spans. See more details in opentracing::Tracer::Close() documentation.
         opentracing::Tracer::Global()->Close();
         return 0;
```

```
ii. Go to the build directory.
```

iii. Run the make command to build an application.

iv. Run the following code.

You must replace the variables in the following code with the actual values. The following table describes the variables.

If you want to print spans, set the required environment variable in the format of export JAEGER_REPORTER_LOG_SPANS=true.

export JAEGER_SAMPLER_TYPE=const export JAEGER SAMPLER PARAM=1

export JAEGER_SERVICE_NAME=\$ {service}

export JAEGER_PROPAGATION=w3c

export JAEGER_ENDPOINT="https://\${endpoint}/jaeger/api/traces"

export JAEGER_TAGS=sls.otel.project=\${project}, sls.otel.instanceid=\${instance}, sls.otel.akid=\${access-key-id}, sls.otel.aksecret=\${access-key-id}, sls.otel.aksecret=\${acces

./app

Table 1. Variables

Variable	Description	Example
\${service}	The name of the service. Specify the value based on your business requirements.	payment
\${version}	The version of the service. We recommend that you specify a version in the $\ensuremath{\textbf{va.b.c}}$ format.	v0.1.2
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations. ⑦ Note • If you set the variable to stdout in the provider.WithTraceExporterEndpoint("stdout"), format, data is printed as standard outputs. • If you leave the variable empty, trace data is not uploaded to Log Service.	trace.test-region-id.sls-pub.test- internet-domain:10010
<i>\${project}</i>	The name of the Log Service project.	test-project
<i>\${instance}</i>	The ID of the trace instance. For more information, seeCreate a trace instance.	test-traces
\${access-key-id}	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 	None
<i>\${access-key-secret}</i>	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None

What to do next

• View the details of a trace instance

• Query and analyze trace data

4.9.1.4.2.10. Import trace data from Android apps to Log Service

This topic describes how to import trace data from Android apps to Log Service by using Log Service SDK for Android.

Prerequisites

A trace instance is created. For more information, see Create a trace instance:

Step 1: Integrate the SDK

You can integrate the SDK in automatic or manual mode.

(Recommended) Integrate the SDK in automatic mode

```
③ Note Only the Maven repository is supported.
```

1. Add the following settings to the build.gradle file of your project:

```
buildscript {
    repositories {
        google()
        jcenter()
        mavenCentral()
    }
}
allprojects {
    repositories {
        google()
        jcenter()
        mavenCentral()
    }
}
```

2. Add the following settings to the build.gradle file of your app:

android { defaultConfig { ndk { // Specify the architecture that is supported by .so libraries. If you do not specify an architecture, all architectures are suppo rted by default abiFilters 'armeabi' //, 'armeabi-v7a', 'arm64-v8a', 'x86', 'x86 64' } } } dependencies { // Use implementation in Gradle 3.0 or later. implementation'com.aliyun.openservices:aliyun-log-android-sdk:2.6.10' implementation'com.aliyun.openservices:sls-android-core:1.0.5' implementation'com.aliyun.openservices:sls-android-ot:1.0.6 implementation'com.aliyun.openservices:sls-android-trace:1.0.3' // If you want to import trace data from the mobile side to the server side, add the following settings and use OkHttp3 as the network 1 ibrary: implementation'com.aliyun.openservices:sls-android-okhttp:1.0.1' } (?) Note If you want to use other network libraries when you import trace data from the mobile side to the server side, contact technical support

Integrate the SDK in manual mode

Download the latest versions of the following SDKs from the Maven repository:

- aliyun-log-android-sdk
- sls-android-core
- sls-android-ot
- sls-android-trace
- sls-android-okhttp

Step 2: Configure permissions

Add the following permission declarations to the AndroidManifest.xml file:

- <uses-permission android:name="android.permission.INTERNET" /> <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>

Step 3: Configure obfuscation

If your project code is packaged and obfuscated, you must add obfuscation settings. Package obfuscation rules require that all the class names and method names of the comaliyun.sls.android package be retained. For example, add the following settings to the progaurd.cfg file:

```
-keep class com.aliyun.sls.android.producer.* { *; }
-keep interface com.aliyun.sls.android.producer.* { *; }
-keep class com.aliyun.sls.android.** { *; }
```

Step 4: Configure access

1. Add an Application class to the \$PROJECT/app/src/main/AndroidManifest.xml file.

The following example shows how to add the MyApplication class:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
         package="com.aliyun.sls.android.demo">
    <application
       android:icon="@mipmap/ic_launcher"
        android:name="com.aliyun.sls.android.demo.SLSDemoApplication"
       android:theme="@style/AppTheme">
    </application>
</manifest>
```

The integrated development environment (IDE) that is used automatically creates a class named MyApplication and adds the class to the current project based on the instructions sent by Android Studio.

2. Add the following initialization code to the MyApplication.onCreate method:

public class MyApplication extends Application {

```
@Override
public void onCreate() {
   super.onCreate();
   Credentials credentials = new Credentials();
   credentials.accessKeyId = "your access key id";
    credentials.accessKeySecret = "your access key secret";
    // If the preceding AccessKey pair is obtained by using Security Token Service (STS), you must configure securityToken.
    credentials.securityToken = "your access security toeken";
    TracerCredentials tracerCredentials = credentials.createTraceCredentials();
   tracerCredentials.endpoint = "your trace endpoint";
tracerCredentials.project = "your trace project";
   tracerCredentials.logstore = "your trace logstore";
    \ensuremath{{//}} Initialize the SLSAndroid class before you use the trace feature.
    SLSAndroid.initialize(
       this,
        credentials,
        configuration -> {
            // Enable the trace feature.
            configuration.enableTracer = true;
        }
    );
```

// If you want to import trace data from the mobile side to the server side, you must specify a network library. For more
information, see the following Step 3.
}

Credentials

The Credentials class defines authentication-related fields.

Field	Example	Description
accessKeyld	LTAI****eYDw	The AccessKey ID that is used to access the required Log Service project. For more information, see the Obtain AccessKey credentials topic in Preparations .
accessKeySecret	lrRq****GOVM	The AccessKey secret that is used to access the required Log Service project. For more information, see the Obtain AccessKey credentials topic in Preparations .
securityToken	124f****a369	The STS token that is used to access the required Log Service project. If you want to use STS to access the project, you must configure this field.

• TracerCredentials

The TracerCredentials class defines important configuration fields.

Field	Example	Description
endpoint	https://trace.test-region- id.sls-pub.test-internet- domain:10010	The endpoint of the project that you specified when you created the trace instance. You must specify <pre>https:// as the prefix of the endpoint. For more information, see theObtain the endpoint of Log Service topic in Preparations.</pre>
project	sls-ayasls-demo	The name of the project that you specified when you created the trace instance. For more information, see Create a trace instance:
logstore	ayasls-traces	The name of the Logstore that is automatically created after you created the trace instance. The name is in the <i>{instance}-traces</i> format, where <i>{instance}</i> is the ID of the trace instance. For more information, see Create a trace instance. You must set this field to the name of the Logstore.

• SLSAndroid

The SLSAndroid class implements the interfaces that are used to configure user information and all parameters related to SDK initialization.

Туре	Field or method	Description
Debugging method	setLogLevel	Configures the log level for the SDK. Valid values: Log.VERBOSE, Log.DEBUG, Log.INFO, Log.WARN, and Log.ERROR.
Cradential undate	setCredentials	Updates credential information. Hot update is supported.
Credential update	registerCredentialsCallback	Calls back the onCall method when span data is sent or failed to be sent.
Configuration method	setUserInfo	Updates user information.
	setExtra	Adds global extension parameters. The operation takes effect globally.
Configuration of extension parameters	removeExtra	Removes global extension parameters. The operation takes effect globally.
	clearExtra	Clears global extension parameters. The operation takes effect globally.

Configuration

The Configuration class implements the interfaces that are used to configure additional parameters other than the parameters required for SDK initialization.

Type Field or method Description	
----------------------------------	--

3.

4.

Configuration method	Configuration mothod	enableTracer	Specifies whether to enable the trace feature.		
	spanProvider	Specifies the resource and attribute information about a custom span.			
			Specifies the environment information about an app. Default value: default.		
	Environment configuration	env	We recommend that you set env to dev for a development environment and to prod for a production environment.		
Op If y ne	Optional:Specify a network library. If you want to import trace data from the mobile side to the server side, you must use OkHttp3 as the network library. To specify OkHttp as the network library, use the following settings at the position where OkHttp is called in your project:				
/ / / c	<pre>// Pass in the client parameter. // We recommend that you use a singleton class to implement the callFactory interface. Call.Factory callFactory = OKHttp3Tracer.newCallFactory(client); // Generate a request. Call call = callFactory.newCall(request);</pre>				
Op	Optional:Call an STS operation to update credentials.accessKeyId, credentials.accessKeySecret, and credentials.securityToken.				
p	public class MyApplication extends Application {				
	<pre>// You can initiate the call after you obtain the information such as the required AccessKey pair. private void onUpdateSLS() { Credentials credentials = new Credentials();</pre>				
	<pre>// (Optional) Update the AccessKey pair. credentials.accessKeyId = "your access key id"; credentials.accessKeySecret = "your access key secret"; credentials.securityToken = "your access security token";</pre>				
	<pre>// (Optional) Update information such as project information. TracerCredentials tracerCredentials = credentials.createTraceCredentials(); tracerCredentials.endpoint = "your trace endpoint"; tracerCredentials.project = "your trace project"; tracerCredentials.logstore = "your trace logstore";</pre>				
}	SLSAndroid.setCredent }	ials(credentials);			

Step 5: Create trace data

Create a single span

The SDK provides multiple methods to create a single span.

Use SpanBuilder

Code format

SpanBuilder builder = Tracer.spanBuilder("span name"); builder.setParent(span); // Specify the parent span. builder.setStart(start); // Specify the start time. builder.addAttribute(attribute); // Add attribute information. builder.addResource(resource); // Add resource information. builder.setActive(true/false); // Specify whether to keep the span active. Span span = builder.build(); // Create and enable a span. span.end(); // Stop the current span.

Configuration example

Span span = Tracer.spanBuilder("span with children (SpanBuilder)")
 .addAttribute(Attribute.of("attr_key", "attr_value"))
 .addResource(Resource.of("res_key", "res_value"))
 .build();
span.end();

Use startSpan

Code format

Span span = Tracer.startSpan("span name"); // After you call the startSpan method, the span is started. span.addResource(resource); // Add attribute information. span.addResource(resource); // Add resource information. span.setStart(start); // Specify the start time. In most cases, you do not need to specify the start time. span.setStart(start); // Specify the start to the span. span.setStatus(ERROR/OK/UNSET); // Specify the status of the span. Default value: UNSET. span.setStatus(ERROR/OK/UNSET); // Specify the start of the description of the span status. span.setStatus("span status message"); // Specify the ID of the parent span. span.setStart("service name"); // Specify the service name. Default value: Android. In most cases, you do not need to specify the service name. span.setSpanId("span id"); // The custom ID of the span. In most cases, you do not need to specify the custom ID of the span. span.setTraceId("trace id"); // The custom ID of the trace. In most cases, you do not need to specify the custom ID of the trace. span.end(); // Stop the current span.

Configuration example

```
Span span = Tracer.startSpan("span 1");
    span.addAttribute(Attribute.of("attr_key", "attr_value"))
        .addResource(Resource.of("res_key", "res_value"));
        span.end();
```

Use withinSpan

Code format

If you use withinSpan to create a span, you do not need to configure the **start** or **end** settings for the span. In most cases, you need to add business code to Runnable. In addition, you cannot specify additional information about the span.

```
Tracer.withinSpan("span name", new Runnable() {
    @Override
    public void run() {
        // The code block.
    }
});
o Configuration example
Tracer.withinSpan("span with block", new Runnable() {
    @Override
    public void run() {
    }
}
```

android.util.Log.d("debug", "print log from withinSpan");
});

Create a piece of trace data that contains multiple spans

A piece of trace data is associated with one or more spans. The SDK provides multiple methods to associate different spans with trace data.

```
• Use SpanBuilder
```

```
Span span = Tracer.spanBuilder("span with children (SpanBuilder)")
    .setActive(true) // Set the value to true.
    .addAttribute(Attribute.of("attr_key", "attr_value"))
    .addResource(Resource.of("res_key", "res_value"))
    .build();
Tracer.startSpan("child span 1 (SpanBuilder)").end();
```

Tracer.startSpan("child span 2 (SpanBuilder)").end();

span.end(); // Stop the trace.

```
• Use startSpan
```

Span span = Tracer.startSpan("span with children", true); // Set the second parameter to true.

Tracer.startSpan("child span 1").end(); Tracer.startSpan("child span 2").end();

span.end(); // Stop the trace.

• Use withinSpan

```
Tracer.withinSpan("span with func block", new Runnable() {
  @Override
  public void run() {
    Tracer.startSpan("span within func block 1").end();
    // The child trace.
    Tracer.withinSpan("nested span with func block", new Runnable() {
      @Override
      public void run() {
         Tracer.startSpan("nested span 1").end();
         Tracer.startSpan("nested span 2").end();
        }
    });
    Tracer.startSpan("span within func block 2").end();
    }
}
```

});

Add custom attribute and resource information

The SDK allows you to use SpanProvider to add custom information. If you use SpanProvider to add custom resource and attribute information, the information is added to all spans. This way, you can add business-related resource and attribute information to all spans at a time.

// The initialization process of the Credentials class is omitted.
//
SLSAndroid.initialize(
this,
credentials,
configuration -> {
// Other settings of the configuration class are omitted.
//
configuration spanProvider = new ISpanProvider() {
Autorida
public Resource provideresource() {
return Resource.of("other_resource_key", "other_resource_value");
}
@Override
<pre>public List<attribute> provideAttribute() {</attribute></pre>
List <attribute> attributes = new ArrayList<>();</attribute>
<pre>attributes.add(Attribute.of("other_attribute_key", "other_attribute_value"));</pre>
return attributes;
}
};
).
// Other configuration processes are omitted
// other configuration processes are omitted.

Parameters

• Tracer

Field or method	Description
spanBuilder(name)	Creates a SpanBuider object.
startSpan(name)	Creates a span object.
startSpan(name, active)	Creates a span object and specifies whether to keep the span active. The value of active determines whether the span is active.
withinSpan(name, runnable)	Creates a span object, keeps the span active, and automatically runs the code block that is specified in runnable.
withinSpan(name, active, runnable)	Creates a span object, specifies whether to keep the span active, and automatically runs the code block that is specified in runnable. The value of active determines whether the span is active.
withinSpan(name, active, parent, runnable)	Creates a span object, specifies whether to keep the span active, specifies the parent span of the span, and automatically runs the code block that is specified in runnable. The value of active determines whether the span is active. The value of parent determines the parent span.

SpanBuilder

Field or method	Description
SpanBuilder(name, processor, provider)	Creates a SpanBuilder object, specifies the name of the span, and configures SpanProcessor and SpanProvider. We recommend that you do not directly call the method.
setParent(parent)	Specifies the parent span.
setActive(active)	Specifies whether to keep the span active.
setKind(SpanKind)	Specifies the type of the span. Valid values: INTERNAL, SERVER, CLIENT, PRODUCER, and CONSUMER. Default value: CLIENT.
setStart(start)	Specifies the start time of the span.
addAttribute(attribute)	Adds attribute information.
addAttribute(attributes)	Adds a set of attribute information.
addResource(resource)	Adds resource information.

• Span

Field or method	Description
setName(name)	Specifies the name of the span.
setKind(SpanKind)	Specifies the type of the span. Valid values: INTERNAL, SERVER, CLIENT, PRODUCER, and CONSUMER. Default value: CLIENT.
setTraceId(traceId)	Specifies the ID of the trace. We recommend that you do not manually specify the ID.
setSpanId(spanId)	Specifies the ID of the span. We recommend that you do not manually specify the ID.
setParentSpanId(spanId)	Specifies the ID of the parent span.
setStart(start)	Specifies the start time of the span. We recommend that you do not manually specify the time.
setEnd(end)	Specifies the end time of the span.

setDuration(duration)	Specifies the duration of the span. We recommend that you do not manually specify the duration.
setStatus(StatusCode)	Specifies the status of the span. Valid values: ERROR, UNSET, and OK. Default value: UNSET.
setStatusMessage(message)	Specifies the description of the span status.
setHost(host)	Specifies the host.
setService(service)	Specifies the service name. Default value: Android.
addAttribute(attribute)	Adds attribute information.
addAttribute(attribute)	Adds attribute information.
addAttribute(attributes)	Adds a set of attribute information.
addResource(resource)	Adds resource information.
end()	Stops the current span.
isEnd()	Checks whether the current span is stopped.
toMap()	Converts the span to a map object.

Resource

Field or method	Description
getDefault()	Returns the default resource object. For more information about the default resource information, see Resource object information.
of(String key, Object value)	Returns a resource object based on the key and value that are passed in.
of(Pair <string, object=""> resources)</string,>	Returns a resource object based on the key-value pair that is passed in.
of(List <attribute> attributes)</attribute>	Returns a resource object based on the attribute list that is passed in.
add(String key, Object value)	Adds key and value information to the current resource object.
merge(Resource resource)	Merges the resource object information that is passed in into the current resource object for implementation.

Table 1. Resource object information

key	value
sdk.language	Android
host.name	Android
device.model.identifier	Build.MODEL
device.model.name	Build.PRODUCT
device.manufacturer	Build.MANUFACTURER
os.type	Linux
os.description	Build.DISPLAY
os.name	Android
os.version	Build.VERSION.RELEASE
os.sdk	Build.VERSION.SDK
host.name	Build.HOST
host.type	Build.TYPE
host.arch	Build.CPU_ABI + (TextUtils.isEmpty(Build.CPU_ABI2) ? "" : (", " + Build.CPU_ABI2))
sls.sdk.version	BuildConfig.VERSION_NAME

• Attribute

Field or method	Description
of(String key, boolean value)	Returns an attribute object.
of(String key, int value)	Returns an attribute object.
of(String key, long value)	Returns an attribute object.
of(String key, double value)	Returns an attribute object.
of(String key, String value)	Returns an attribute object.
of(final String key, final Object value)	Returns an attribute object.
of(Pair <string, object=""> kvs)</string,>	Returns an attribute list object.

Additional information

If you set a span to active, the new spans in the current context are all automatically associated with the active span. Implementations: currentSpan.parentSpanID = activeSpan.spanID and currentSpan.traceID = activeSpan.traceID

ONOTE If the code of different services runs on the same thread, the thread is the current context. Only one span can be active in the same context

What to do next

- View the details of a trace instance
- Ouery and analyze trace data

4.9.1.4.2.11. Import trace data from iOS apps to Log Service

This topic describes how to import trace data from iOS apps to Log Service by using Log Service SDK for iOS.

Prerequisites

A trace instance is created. For more information, see Create a trace instance.

Step 1: Integrate the SDK

(Recommended) Use CocoaPods

```
1. Add the following content to a Podfile in your Xcode project:
```

// Add the aliyun-specs source. source'https://github.com/aliyun/aliyun-specs.git' // Add the CocoaPods source. source'https://github.com/CocoaPods/Specs.git' pod'AliyunLogProducer', '~>3.1.6', :subspecs=>['Trace']

2. Save the changes and execute the pod install --repo-update command.

3. Use the file that is suffixed with .xcworkspace to open your project.

Use an SDK file

1. Download the SDK file.

- Decompress the SDK file and add the following framework files to your project: AliyunLogProducer.framework, AliyunLogCore.framework, AliyunLogOT.framework, and AliyunLogTrace.framework

Step 3: Generate trace data

Create a single span

The SDK provides multiple methods to create a single span.

Use SpanBuilder

```
SLSSpan *span = [[[[[[[SLSTracer spanBuilder:@"span with spanbuilder"]
                   setParent:span] // Set the parent span.
                   setStart:start] // Set the start time.
                  setActive:YES/NO] // Set whether to keep the span active.
                   setService: @"service name"] // Set the service name. Default value: iOS. In most cases, you do not need to set the servic
e name.
                  addAttribute:[SLSAttribute of:@"attr_key" value:@"attr_value"], nil] // Add attribute information.
                   addResource:[SLSResource of:@"res key" value:@"res value"]] // Add resource information.
                build]; // Create and start a span.
[span end]; // End the current span.
```

• Use startSpan

SLSSpan *span = [SLSTracer startSpan:@"span name"]; // After you call the startSpan method, the span is started.

- [span setName: @"span name"]; // Set the name of the span.
- [span setParentSpanID:parent.spanID]; // Set the ID of the parent span.
- [span setStart:start]; // Set the start time. In most cases, you do not need to set the start time.
- [span setService:@"service name"]; // Set the service name. Default value: iOS. In most cases, you do not need to set the service name.
- [span setStatusCode:ERROR/UNSET/OK]; // Set the status of the span. Default value: UNSET.
- [span setStatusMessage:@"status message"]; // Set the description of the span status.
- [span setKind:SLSINTERNAL/SLSSERVER/SLSCLIENT/SLSPRODUCER/SLSCONSUMER];
- [span addAttribute:[SLSAttribute of:@"attr_key" value:@"attr_value"], nil]; // Add attribute information. [span addResource:[SLSResource of:@"res_key" value:@"res_value"]]; // Add resource information.
- [span end]; // End the current span.

• Use withinSpan

If you use withinSpan to create a span, you do not need to configure the **start** and **end** settings for the span. In most cases, you need to add business code to the Runnable interface. In addition, you cannot specify additional information about the span.

[SLSTracer withinSpan:@"span with func block" block:^{ // The code block.

}];

Create a piece of trace data that contains multiple spans

A piece of trace data is associated with one or more spans. The SDK provides multiple methods to associate different spans with trace data.

Use SpanBuilder

[[Tracer startSpan: @"child span 2"] end];

[span end]; // End the trace.

• Use startSpan

SLSSpan *span = [SLSTracer startSpan:@"span with children" active:YES]; // Set active to YES.
[[SLSTracer startSpan:@"child span 1"] end];
[[SLSTracer startSpan:@"child span 2"] end];
[span end]; // End the trace.

```
• Use withinSpan
```

```
[SLSTracer withinSpan:@"span with func block" block:^{
   [[SLSTracer startSpan:@"span within block 1"] end];
   // The child trace.
   [SLSTracer withinSpan:@"nested span with func block" block:^{
      [[SLSTracer startSpan:@"nested span 1"] end];
      [[SLSTracer startSpan:@"nested span 2"] end];
   }];
   [[SLSTracer startSpan:@"span within block 2"] end];
}];
```

Add custom attribute and resource information

The SDK allows you to use SpanProvider to add custom information. If you use SpanProvider to add custom resource and attribute information, the information is added to all spans. This way, you can add business-related resource and attribute information to all spans at a time.

```
// Implement SLSSpanProviderProtocol.
 @interface SpanProvider : NSObject<SLSSpanProviderProtocol>
 + (instancetype) provider;
 @end
 @implementation SpanProvider
 + (instancetype) provider {
    return [[SpanProvider alloc] init];
 }
 - (nonnull NSArray<SLSAttribute *> *)provideAttribute {
    NSMutableArray<SLSAttribute *> *array = [NSMutableArray array];
     [array addObject:[SLSAttribute of:@"attr_key" value:@"attr_value"]];
     return array;
 }
 - (nonnull SLSResource *)provideResource {
    return [SLSResource of:@"res key" value:@"res value"];
 @end
 // The initialization process of credentials is omitted.
 // ...
 [[SLSCocoa sharedInstance] initialize:credentials configuration:^(SLSConfiguration * _Nonnull configuration) {
    // Other configuration settings are omitted.
     configuration.spanProvider = [SpanProvider provider];
 }];
 \ensuremath{{\prime}}\xspace // Other configuration processes are omitted.
 // ...
Parameters
```

```
    SLSTracer
```

Field or method	Description
spanBuilder:	Constructs an SLSSpanBuilder object.
startSpan:	Constructs an SLSSpan object.
startSpan: active:	Constructs an SLSSpan object and sets active. The value of active determines whether the span is active.
withinSpan: block:	Constructs an SLSSpan object, sets the span to active, and automatically runs the code of block.
withinSpan: active: block:	Constructs an SLSSpan object, sets active, and automatically runs the code of block. The value of active determines whether the span is active.
withinSpan: active: parent: block:	Constructs an SLSSpan object, sets active, sets parent, and automatically runs the code of block. The value of active determines whether the span is active. The value of parent determines the parent span.

SLSSpanBuilder

Field or method	Description
builder	Constructs an SLSSpanBuilder object.
initWithName: provider: processor:	Initializes the SLSSpanBuilder object, sets the name of the span, and configures SpanProcessor and SpanProvider.
setParent:	Sets the parent span.
setActive:	Sets whether the span is active.
setKind:	Sets the type of the span. Valid values: SLSINTERNAL, SLSSERVER, SLSCLIENT, SLSPRODUCER, and SLSCONSUMER. Default value: SLSCLIENT.
setStart:	Sets the start time of the span.
addAttribute: ,	Adds a set of attribute information.
addAttributes:	Adds a set of attribute information.
addResource:	Adds resource information.
setService:	Sets the service name. Default value: iOS.

SLSSpan

Field or method	Description
setName:	Sets the name of the span.
setKind:	Sets the type of the span. Valid values: INTERNAL, SERVER, CLIENT, PRODUCER, and CONSUMER. Default value: CLIENT.
setTraceId:	Sets the ID of the trace. We recommend that you do not manually set the ID.
setSpanId:	Sets the ID of the span. We recommend that you do not manually set the ID.
setParentSpanId:	Sets the ID of the parent span.
setStart:	Sets the start time of the span. We recommend that you do not manually set the time.
setEnd:	Sets the end time of the span.
setDuration:	Sets the duration of the span. We recommend that you do not manually set the duration.
setStatus:	Sets the status of the span. Valid values: ERROR, UNSET, and OK. Default value: UNSET.
setStatusMessage:	Sets the description of the span status.
setHost:	Sets the host.
setService:	Sets the service name. Default value: iOS.
addAttribute: ,	Adds a set of attribute information.
addAttributes:	Adds a set of attribute information.
addResource(resource)	Adds resource information.
end	Ends the current span.
isEnd	Checks whether the current span is ended.
toDict	Converts the span to a map object.

SLSResource

Field or method	Description
resource	Returns an SLSResource object.
of: value:	Returns a resource object based on the input key and value.
of: ,	Returns a resource object based on the input key-value pair of SLSKeyValue.
ofAttributes:	Returns a resource object based on the input attribute array information.
add: value:	Adds key and value information to the current resource object.
add:	Adds a set of SLSAttribute information to the current resource object.
merge:	Merges the input information of the resource object into the current resource object for implementation.

• Attribute

Field or method	Description
of: value:	Returns an attribute object.
of: ,	Returns an attribute array object.

Additional information

If you set a span to active, the new spans in the current context are all automatically associated with the active span. Implementations:

currentSpan.parentSpanID = activeSpan.spanID and currentSpan.traceID = activeSpan.traceID .

③ Note If the code of different services runs on the same thread, the thread is the current context. Only one span can be active in the same context.

What to do next

- View the details of a trace instance
- Query and analyze trace data

4.9.1.4.3. Existing import methods

4.9.1.4.3.1. Import trace data from OpenCensus to Log Service

You can send trace data from OpenCensus to the OpenTelemetry Collector by using OpenCensus SDK, and then forward the data to Log Service by using the OpenTelemetry Collector. This topic describes how to forward trace data to Log Service by using the OpenTelemetry Collector.

Prerequisites

A trace instance is created. For more information about the networking of cloud boxes, see Create a trace instance.

Procedure

- 1. Install the OpenTelemetry Collector.
- i. Download the OpenTelemetry Collector.
- ii. Configure the OpenTelemetry Collector.
- a. Create a file named config.yaml.
- b. Add the following code to the config.yaml file.

Replace the variables in the following code with the actual values. For more information about the variables, see Variables.

receivers:
opencensus:
endpoint: 0.0.0.0:6850
exporters:
logging/detail:
loglevel: debug
alibabacloud_logservice/traces:
endpoint: "\${endpoint}"
project: "\${project}"
logstore: "\${instance}-traces"
access_key_id: "\${access-key-id}"
access_key_secret: "\${access-key-secret}"
service:
pipelines:

traces: receivers: [opencensus] # Set the value to opencensus. exporters: [alibabacloud_logservice/traces] # Set the value to alibabacloud_logservice/traces. # for debug #exporters: [logging/detail,alibabacloud_logservice/traces]

Table 1. Variables

Variable	Description	Example
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations.	trace.test-region-id.sls-pub.test- internet-domain:10010
<i>\${project}</i>	The name of the Log Service project.	test-project
<i>\${instance}</i>	The ID of the trace instance. For more information, seeCreate a trace instance.	test-traces
\${access-key-id}	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 	None
\${access-key-secret}	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None

iii. Start the OpenTelemetry Collector.

./otelcontribcol_linux_amd64 --config="./config.yaml"

2. Configure OpenCensus.

Change the endpoint of OpenCensus to the endpoint on which the OpenTelemetry Collector listens. For example, if the endpoint of the OpenTelemetry Collector is *\${collector-host}*, you must set the endpoint of OpenCensus to *\${collector-host}*:6850.

What to do next

• View the details of a trace instance

• Query and analyze trace data

4.9.1.4.3.2. Import trace data from Zipkin to Log Service

You can import trace data from Zipkin to Log Service. You can also use the OpenTelemetry Collector to forward trace data to Log Service.

Prerequisites

A trace instance is created. For more information, see Create a trace instance:

Import trace data from Zipkin

If you want to use the Zipkin protocol to import trace data to Log Service, you must configure the endpoint and authentication settings in the Zipkin SDK.

▲ Warning To ensure data security during transmission, you must import data over HTTPS.

- Endpoint settings
- HTTP 2.0: An HTTPS endpoint is in the \${endpoint}/zipkin/api/v2/spans format. Example: https://test-project.cn-hangzhouintranet.log.aliyuncs.com/zipkin/api/v2/spans. We recommend that you use this type of endpoint.
- HTTP 1.0: An HTTPS endpoint is in the \${endpoint}/zipkin/api/v1/spans format. Example: https://test-project.cn-hangzhou.log.aliyuncs.com/zipkin/api/v1/spans.

You must replace the *\${endpoint}* variable with the actual value. The following table describes the variable. Table 1. Variables

Variable	Description	Example
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations.	trace.test-region-id.sls-pub.test- internet-domain:10010

Authentication settings

You can configure the authentication settings in HTTPS header fields. The following table describes the fields.

HTTPS Header Key	Description	Example
x-sls-otel-project	The name of the Log Service project.	test-project
x-sls-otel-instance-id	The ID of the trace instance. For more information, seeCreate a trace instance:	test-traces
x-sls-otel-ak-id	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. o For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 	None
x-sls-otel-ak-secret	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None

Use the OpenTelemetry Collector to forward trace data

You can use the Zipkin SDK to import trace data from Zipkin to the OpenTelemetry Collector, and then use the OpenTelemetry Collector to forward the data to Log Service. This method supports data transmission over HTTP or HTTPS.

1. Install the OpenTelemetry Collector.

i. Download the OpenTelemetry Collector.

ii. Configure the OpenTelemetry Collector.

- a. Create a file named config.yaml.
- b. Add the following code to the config.yaml file.
 - Replace the variables in the following code with the actual values. For more information about the variables, see Variables.

receivers:
zipkin:
endpoint: 0.0.0.0:9411
exporters:
logging/detail:
loglevel: debug
alibabacloud_logservice/traces:
endpoint: "\${endpoint}"
project: "\${project}"
logstore: "\${instance}-traces"
access_key_id: "\${access-key-id}"
access_key_secret: "\${access-key-secret}"
service:
pipelines:
traces:
receivers: [zipkin] # Set the receivers parameter to zipkin.
exporters: [alibabacloud_logservice/traces] # Set the value to alibabacloud_logservice/sls-traces.

for debug
#exporters: [logging/detail,alibabacloud_logservice/traces]

Table 2. Variables

Variable	Description	Example
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations.	trace.test-region-id.sls-pub.test- internet-domain:10010
<i>\${project}</i>	The name of the Log Service project.	test-project
<i>\${instance}</i>	The ID of the trace instance. For more information, seeCreate a trace instance:	test-traces
\${access-key-id}	The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	
	 For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. 	None
	 For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 	
<i>\${access-key-secret}</i>	The AccessKey secret of your Alibaba Cloud account.	
	We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None

iii. Start the OpenTelemetry Collector.

./otelcontribcol_linux_amd64 --config="./config.yaml"

2. Configure Zipkin.

Change the output endpoint of Zipkin to an endpoint on which the OpenTelemetry Collector can listen. For example, if the endpoint of the OpenTelemetry Collector is *\${collector-host}*, change the output endpoint of Zipkin to *\${collector-host}*:9411.

What to do next

- View the details of a trace instance
- Query and analyze trace data

4.9.1.4.3.3. Import trace data from Apache SkyWalking to Log Service

This topic describes how to import trace data from Apache SkyWalking to Log Service. This way, you can query and analyze the trace data by using Log Service.

Prerequisites

- Apache SkyWalking
- A SkyWalking agent of version 8.0.0 or later is installed in the application on which data is collected. For more information, see Setup.

Log Service

- A Logstore that stores trace data and a Metricstore that stores time series data are created. For more information, see Create a Logstore and Create a Metricstore.
- A machine group is created, and the group uses a custom identifier. For more information, see Create a custom identifier-based machine group.
 - () Important Make sure that the custom identifier is unique in the region of the Log Service project to which the Logstore belongs.

Background information

We recommend that you import trace data from Apache SkyWalking to Log Service. This brings the following benefits:

- Elasticity: Log Service can handle traffic spikes in an efficient manner.
- Performance: Log Service provides higher query performance than open source Elasticsearch. Log Service allows you to write petabytes of data per day, and returns results to queries for billions or tens of billions of data rows within seconds.
- Stability: Log Service uses three replicas for storage, which provides high availability and reliability.

• O&M: Log Service provides an out-of-the-box feature that allows you to import Apache SkyWalking trace data. O&M is not required for Log Service. You do not need to perform O&M on servers or applications.

Procedure

The following procedure describes how to import trace data from Apache SkyWalking to Log Service:

- 1. Create a Logtail configuration.
- i. Log on to the Log Service console
- ii. In the Import Data section, select SkyWalking.
- Click View More. In the Import Data dialog box, enter SkyWalking.
- iii. In the Specify Logstore step, select the project and Logstore. Then, click Next.
- iv. In the Create Machine Group step, click Use Existing Machine Groups
- v. In the Machine Group Settings step, select the machine group that you want to use in the Source Server Groups section and move the machine group to the Applied Server Groups section. Then, click Next.
- vi. In the Specify Data Source step, configure the Logtail plug-ins and click Next. A configuration template is provided in the Plug-in Config field. Replace \${logstorename} with the Logstore name and \${metricstorename} with
- the Metricstore name.

? Note If the local port 11800 of your Logtail is occupied, you can use another available port. In this case, you must change the port that is used by Apache SkyWalking to send data.

```
"inputs" : [
   {
       "detail" : {
           "Address" : "0.0.0.0:11800"
       },
        "type" : "service_skywalking_agent_v3"
   }
"aggregators" : [
   {
       "detail" : {
           "MetricsLogstore" : "${logstorename}",
           "TraceLogstore" : "${metricstorename}"
       }.
       "type" : "aggregator skywalking"
   }
"global" : {
   "AlwaysOnline" : true,
   "DelayStopSec" : 300
```

- 2. Log on to the server where your Java application is uploaded, modify the configurations, and then start the SkyWalking service.

 - i. Run the **vim** command to modify the port number in the webapp.yml configuration file. To prevent port conflicts, make sure that the port number in the file is different from that of the Java application. Sample command:

vim apache-skywalking-apm-bin/webapp/webapp.yml

ii. Go to the bin directory of SkyWalking and execute the ./startup.sh script to start the service. If the following message appears, the service is started:

SkyWalking OAP started successfully!

3. Start the lava application. Run the following command:

java -javaagent:/home/admin/skywalking-agent/skywalking-agent.jar -Dskywalking.agent.service_name=skywalking-java-demo -Xmx512M -jar skywalk ing-java-demo.jar &

If the following message appears, the application is started:

Tomcat started on port(s): 8080 (http)

4. Run the following command to call the Java application and use Logtail to collect logs.

Start the Java application and use Logtail to collect logs based on your business requirements. Sample command:

curl localhost:8080/hello-world

If the following message appears, the application is started:

hello world

5. Return to the Log Service console and click Next. After you complete the Logtail configuration, Log Service starts to collect data.

? Note

- A Logtail configuration requires up to 3 minutes to take effect.
- If an error occurs when you use Logtail to collect logs, see How do I view Logtail collection errors?.

What to do next

After you import trace data from Apache SkyWalking to Log Service, you can perform the following operations:

Query and analyze the log data of traces

· Query and analyze the time series data of traces

4.9.1.4.3.4. Import trace data from OpenTelemetry to Log Service

You can import trace data from OpenTelemetry to Log Service, or forward the trace data to Log Service by using the OpenTelemetry Collector.

Prerequisites

A trace instance is created. For more information, see Create a trace instance.

Import trace data from OpenTelemetry

When you use the OpenTelemetry protocol to import trace data to Log Service, you must configure the endpoint and authentication settings in OpenTelemetry. The following examples describe the required settings:

- Endpoint settings
 - An HTTPS endpoint is in the \${endpoint}/opentelemetry/v1/traces format. Example: http://trace.test-region-id.sls-pub.test-internetdomain/opentelemetry/v1/traces.
 - A gRPC endpoint is in the \${endpoint}:10010 format. Example:trace.test-region-id.sls-pub.test-internet-domain:10010.

Warning If you want to ensure data security during transmission, you must enable Transport Layer Security (TLS) when you use the gRPC protocol.

You must replace the \${endpoint} variable with the actual value. The following table describes the variable. Table 1. Variables

Variable	Description	Example
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations.	trace.test-region-id.sls-pub.test- internet-domain:10010

Authentication settings

You can configure the authentication settings in the header of the gRPC or HTTPS protocol, or in the Resource field of the OpenTelemetry protocol. The following table describes the required fields.

OpenTelemetry Resource	gRPC/HTTPS Header Key	Description	Example
sls.otel.project	x-sls-otel-project	The name of the Log Service project.	test-project
sls.otel.instanceid	x-sls-otel-instance-id	The ID of the trace instance. For more information, seeCreate a trace instance.	test-otel
sls.otel.akid	x-sls-otel-ak-id	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see the Obtain AccessKey credentials topic in Preparations. 	None
sls.otel.aksecret	x-sls-otel-ak-secret	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None

Forward trace data by using the OpenTelemetry Collector

1. Download the OpenTelemetry Collector.

- 2. Configure the OpenTelemetry Collector.
 - i. Create a file named config.yaml.

ii. Add the following code to the config.yaml file. Replace the variables in the following code with the actual values. For more information about the variables, see Variables.

receivers:
otlp:
protocols:
grpc:
endpoint: "0.0.0.0:55680"
http:
endpoint: "0.0.0.0:55681"
exporters:
logging/detail:
loglevel: debug
alibabacloud_logservice/traces:
endpoint: "\${endpoint}"
project: "\${project}"
logstore: "\${instance-id}-traces"
access_key_id: "\${access-key-id}"
access_key_secret: "\${access-key-secret}"
alibabacloud_logservice/metrics:
endpoint: "\${endpoint}"
project: "\${project}"
logstore: "\${instance-id}-metrics"
access_key_id: "\${access-key-id}"
access_key_secret: "\${access-key-secret}"
alibabacloud_logservice/logs:
endpoint: "\${endpoint}"
project: "\${project}"
logstore: "\${instance-id}-logs"
access_key_id: "\${access-key-id}"
access_key_secret: "\${access-key-secret}"
service:
pipelines:
traces:
receivers: [otlp] # Set the value to otlp.
exporters: [alibabacloud_logservice/traces] # Set the value to alibabacloud_logservice/traces.
for debug
<pre>#exporters: [logging/detail,alibabacloud_logservice/traces]</pre>
metrics:
receivers: [otlp]
exporters: [alibabacloud_logservice/metrics]
logs:
receivers: [otlp]
sumantana, (slitebas)and lassanias(lass)

Table 2. Variables

Variable	Description	Example
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. For more information, see the Obtain the endpoint of Log Service topic in Developer Guide.	trace.test-region-id.sls-pub.test- internet-domain:10010
\${project}	The name of the Log Service project.	test-project
\${instance-id}	The ID of the trace instance. For more information, seeCreate a trace instance.	test-traces
\${access-key-id}	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 	None
\${access-key-secret}	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None

3. Start the OpenTelemetry Collector.

./otelcontribcol_linux_amd64 --config="./config.yaml"

What to do next

- View the details of a trace instance
- Query and analyze trace data

4.9.1.4.3.5. Import trace data from Jaeger to Log Service

You can import trace data from Jaeger to Log Service, or forward the trace data to Log Service by using the OpenTelemetry Collector.

Prerequisites

A trace instance is created. For more information, see Create a trace instance.

Import trace data from Jaeger

If you use the Jaeger protocol to import trace data to Log Service, you must configure the endpoint and authentication settings in the Jaeger tracing platform. The following list describes the required settings:

- Endpoint settings
 - The HTTPS endpoint is in the \${endpoint}/jaeger/api/traces format. Example: http://trace.\${region_id}.sls-pub.\${internet-domain}/jaeger/api/traces.
 - A gRPC endpoint is in the **\${endpoint}:10010** format. Example: trace.\${region_id}.sls-pub.\${internet-domain}:10010.
 - **Warning** If you want to ensure data security during transmission, you must enable Transport Layer Security (TLS) when you use the gRPC protocol.

You must replace the *\${endpoint}* variable with the actual value. The following table describes the variable. Table 1. Variables

Variable	Description	Example
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations.	trace.test-region-id.sls-pub.test- internet-domain:10010

Authentication settings

You can configure the authentication settings in the header of the gRPC or HTTPS protocol, or in the Tag field of the Jaeger protocol. The following table describes the required fields.

Jaeger Tag	gRPC/HTTPS Header Key	Description	Example
sls.otel.project	x-sls-otel-project	The name of the Log Service project.	test-project
sls.otel.instanceid	x-sls-otel-instance-id	The ID of the trace instance. For more information, see Create a trace instance.	test-traces
sls.otel.akid	x-sls-otel-ak-id	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see the Obtain AccessKey credentials topic in Preparations. 	None
sls.otel.aksecret	x-sls-otel-ak-secret	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None

Forward trace data to ARMS by using OpenTelemetry Collector

1. Install the OpenTelemetry Collector.

i. Download the OpenTelemetry Collector.

ii. Configure the OpenTelemetry Collector.

- a. Create a file named config.yaml.
- b. Add the following code to the config.yaml file.
 - Replace the variables in the following code with the actual values. For more information about the variables, see Variables.

receivers:
jaeger:
protocols:
grpc:
endpoint: 0.0.0.0:6831
thrift_binary:
endpoint: 0.0.0.0:6832
thrift_compact:
endpoint: 0.0.0.0:6833
thrift_http:
endpoint: 0.0.0.0:6834
exporters:
logging/detail:
loglevel: debug
alibabacloud_logservice/sls-traces:
endpoint: "\${endpoint}"
project: "\${project}"
logstore: "\${instance}-traces"
access_key_id: "\${access-key-id}"
access_key_secret: "\${access-key-secret}"
service:
pipelines:
traces:
receivers: [jaeger]
[alibabacloud_logservice/sls-trace] # Set the value to alibabacloud_logservice/sls-trace.
for debug
#exporters: [logging/detail.alibabacloud logservice/sls-traces]

Table 2. Variables

Variable	Description	Example
\${endpoint}	The endpoint of the Log Service project. Format:trace.\${region_id}.sls- pub.\${internet-domain}:10010. For more information, see the Obtain the endpoint of Log Service topic in Preparations.	trace.test-region-id.sls-pub.test- internet-domain:10010
\${project}	The name of the Log Service project.	test-project
<i>\${instance}</i>	The ID of the trace instance. For more information, seeCreate a trace instance.	test-traces
\${access-key-id}	 The AccessKey ID of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. For information about how to grant a RAM user the write permissions on the Log Service project, see RAM management. For information about how to obtain an AccessKey pair, see theObtain AccessKey credentials topic in Preparations. 	None
\${access-key-secret}	The AccessKey secret of your Alibaba Cloud account. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project.	None

iii. Start the OpenTelemetry Collector.

./otelcontribcol_linux_amd64 --config="./config.yaml"

2. Configure Jaeger.

Change the output endpoint of the Jaeger tracing system to an endpoint on which the OpenTelemetry Collector can listen. For example, if the endpoint of the OpenTelemetry Collector is *\${collector-host}*, you must change the output endpoint of the Jaeger tracing system to *\${collector-host}*; *\${port}*/api/traces.

O Note If an error is returned by the OpenTelemetry Collector because data fails to be parsed, you can switch between the four receiving modes of the Jaeger tracing system to troubleshoot the issue.

What to do next

- View the details of a trace instance
- Query and analyze trace data

4.9.1.4.4. View the import results of trace data

After you import trace data to Log Service, you can query the related logs to check whether trace data is imported to Log Service.

Prerequisites

Trace data is collected. For more information, see Import trace data.

Procedure

- 1. Log on to the Log Service console.
- 2. Open the Custom Query page.

- i. In the Log Application section, click Trace
- ii. In the trace instance list, click the specified instance.
- iii. In the left-side navigation pane, click **Custom Query**.
- 3. Enter a query statement.
 - For example, to check whether trace data that is related to the user service is imported, execute the following query statement:

service:user

4. View the query result.

If the query result contains logs of the user service, you have imported the trace data that is related to the user service.

4.9.1.5. View the details of a trace instance

After you import trace data to Log Service, you can view the service details, trace details, and trace dashboards on the details page of a trace instance.

Prerequisites

Trace data is collected. For more information, see Import trace data.

View service details

After you import the trace data of an application to Log Service, you can view all services and service calls of the application in the service list. You can view information such as QPS, error rate, average latency, and P50 latency. You can also filter services and service operations by time range. For example, if you want to monitor the status of a marketplace system, you can import the trace data of the system to Log Service. All services and service calls of the system are displayed in the service list. The services include user, order, frontend, queue-master, shipping, and job-server.

View trace details

On the Trace Details page, you can view trace details, including the trace trail map and span data. For more information, see View trace details.

Query and analyze trace data

On the **Trace Analysis** page, you can set query conditions to filter trace data. You can also obtain statistics by service group. For more information, see Query and analyze trace data.

View the topological relationship of services

The topology of services shows the dependencies among these services. You can view the topological relationships among these services on the **Trace Analysis > Dependency Analysis** tab or on the **Topology Query** page.

• After you select a service, click the Aggregate icon. The system shows only the services that depend on the selected service.

Click a service. The system shows the average latency, error rate, and number of spans of the service.

Query logs

Log Service records the logs that are generated when you import trace data. You can query and analyze the logs on the Search & Analysis page. For more information, see Query and analyze logs.

View dashboards

By default, Log Service provides two dashboards. The following table describes the dashboards.

Dashboard	Description
Import Overview	Displays the analysis results of trace data. The results include the number of spans, average latency, services with top 10 latency, services with top 10 requests, and top latency methods.
	Log on to the Irace instance. In the left-side navigation pane, choose Overview > Import Overview to view the dashboard.
	Displays the analysis results of trace data. The results include the number of spans, average latency, services with top 10 latency, and services with top 10 requests.
Statistics	Log on to the Trace instance. In the left-side navigation pane, choose Management > Statistics to view the dashboard.

View storage settings

After you create a trace instance and import trace data to Log Service, Log Service generates a Logstore and a Metricstore to store the trace data, for example, {instance}-traces Logstore and {instance}-metrics Metricstore. For more information, see Assets.

In the Storage Setting section, you can view the property of the Logstore or Metricstore.

4.9.1.6. Query and analyze trace data

After you import trace data, you can query the trace data and set statistics by group.

Prerequisites

Trace data is collected. For more information, see Import trace data.

Procedure

- 1. Log on to the Log Service console.
- 2. Open the Trace Analysis page.
 - i. In the Log Application section, click Trace.
- ii. In the trace instance list, click the specified instance.
- iii. In the left-side navigation pane, click Trace Analysis.
- 3. Set query conditions, select a time range, and then click **Search & Analyze**.
- Log Service provides preset query conditions for multiple fields, such as Service, Operation, Duration, Status, Attribute, and Resource. You only need to select these query conditions based on your business requirements. For more information, see Trace data formats.

Cloud Defined Storage

Note

- The value of the Duration parameter in the query condition is measured in milliseconds. However, the unit of the condition expression is microseconds. For example, if you set the Duration parameter to 10 ms, this value is displayed as duration >= 10000 in the filter condition.
- $\circ~$ By default, the value of the Duration parameter is a left-closed and right-open interval.
- The Attribute and Resource fields are of the JSON object type. Log Service allows you to filter the fields by key and value in this field.

For example, you want to query the trace data of the last hour for the user service that has a latency of more than 10 ms. You can set the filter conditions shown in the following figure.

Trace Analysis				
1 (service	e : "user") and du	ration >= 10000	1 Hour(Relative)	Search & Analyze
Common Q	uery Advanced	Query	Туре	Average Latency 🗸
Service	IN V	User ×		140ms 120ms 2.4% 100ms 2.3%
Operation	IN \vee	Please Select V		80ms 2.2% 2.1%
Duration	BETWEEN	10 ms — ms	10:45	
Status	IN \vee	Please Select V		
Attributes				
+ Add Attribute	Filter			
		Search Reset		

4. On the Trace Analysis tab, view the query results.

Trace Analysis Dependency Analysis			
Statistics by Group			
Service \$	Operation \$	Duration \$	Start Time 🗘
user	GET /customers	19.52 ms	2021-06-02 10:52:15
user	GET /customers	10.6 ms	2021-06-02 10:52:14
user	GET /customers	12.64 ms	2021-06-02 10:52:14
user	GET /cards	11.62 ms	2021-06-02 10:52:13

5. Set statistics by group

- i. On the Trace Analysis tab, click Statistics by Group.
- ii. Select a condition for the statistics. In this example, select service.

iii. View the statistical results. Log Service lists the information of each service by service, for example, the number of spans, queries per second (QPS), and average latency.

Trace Analysis	Dependency Analysis								
Statistics by Gro	oup: service ×								
service \$	Spans ¢	QPS \$	Average Latency \$	P50Latency \$	P90Latency 🗘	P95Latency ‡	P99Latency \$	Error Rate ≑	l
user	3945	1.09583333333333334	16.62 ms	14.93 ms	24.16 ms	28.89 ms	41.1 ms	0%	I

4.9.1.7. View trace details

After you import trace data to Log Service, you can view the trace data, including the trace map and span data.

Prerequisites

Trace data is collected. For more information, see Import trace data.

Procedure

- 1. Log on to the Log Service console.
- 2. Open the Trace Analysis page.
 - i. In the Log Application section, click Trace.
- ii. In the trace instance list, click the specified instance.
- iii. In the left-side navigation pane, click Trace Analysis.
- 3. On the Trace Analysis tab, click the specific trace data.
- 4. View the details of the trace.

Cloud Defined Storage

User Guide-Log Service

Service	orders			1 Hour(Relative) 🔻
Enter a keyword				ava 250 200 255 2
Eliter a Reyfford.	Operations	Filter	Q. 1	avg poo poo poo poo poo poo poo poo poo po
shipping front-end	BasicErrorController.error Average Lat Error Ra ency e Ous 0%	QPS 00ops/s		Latency 6s N A A
user queue-master	HTTP POST Average Lat ency 101.26ms	QPS 0.7ops/s	hourse	
	("buildinfo": "?") Average Lat ency 275.31ms	QPS 0.6ops/s	Mm Mm	0939 0948 0957 10.06 10.15 10.24 10.33 Error Rate
	ApplicationDispatcher.forward Average Lat Error Ra ency e Ous 0%	QPS 0.0ops/s		
	/health Error Ra Average Lat error Ra ency e 277.13ms 0%	QPS 0.6ops/s	Mm Mm	0%09:39 09:48 09:57 10:06 10:15 10:24 10:33
	OrdersController.newOrder			QPS
No.	Description			
	 The trace map shows Different colors re 	s the distribution of trace present different service	s and spans. Ta s. In this exampl	ike note of the following information: le, the blue color represents the front-end service. e time that is consumed by a span. The time consumed by a
1	 The length of the I span is calculated consumed by child this case, Span A t The timeline repre 	based on the following fol	che trace.	nsumed by a span = Time consumed by all spans - Time an A takes 10 milliseconds and Span B takes 8 milliseconds. In
2	 Ine length of the l span is calculated consumed by child this case, Span A I The timeline repre Each row in this sect span. Each parent sp can perform the follo Click the Collapse Select a span and Click the Defocus i 	based on the following fi I spans. Assume Span A akes 2 milliseconds. sents the time range of I ion represents a span an an is preceded by a num wing operations in this s icon to collapse or expar click the Aggregate icon con to defocus a span.	represented in time co calls Span B. Spi the trace. d shows the hiel ober, which indic ection: id a span. The system dis	nsumed by a span = Time consumed by all spans - Time an A takes 10 milliseconds and Span B takes 8 milliseconds. In erarchical relationship between the parent span and the child cates the number of child spans owned by the parent span. You splays only the data of the span.

4.9.1.8. Best practices

4.9.1.8.1. Import trace data from Log Service to Grafana

Grafana provides a comprehensive user interface (UI). This topic describes how to import trace data from Log Service to Grafana for visualized analysis.

Prerequisites

- Grafana 8.0.0 or a later version is installed. For more information, see Install Grafana.
- (?) Note In this topic, Grafana 8.0.6 that runs on a Linux operating system is used as an example.
- Trace data is collected. For more information, see Overview.
- The project package that contains the Log Service plug-in is downloaded.

Step 1: Install the Log Service plug-in

The following procedure describes how to install the Log Service plug-in for Grafana.

- 1. Run the following commands to decompress the project package to the plug-in directory of Grafana.
 - If Grafana is installed by using a YUM repository or an RPM package, run the following command:

unzip aliyun-log-grafana-datasource-plugin-master.zip -d /var/lib/grafana/plugins

- $\circ~$ If Grafana is installed by using a .tar.gz file, run the following command:
- {PATH_TO} specifies the installation directory of Grafana.

unzip aliyun-log-grafana-datasource-plugin-master.zip -d {PATH_TO}/grafana-8.0.6/data/plugins

2. Modify the configuration file of Grafana.

- i. Open the configuration file.
 - If Grafana is installed by using a YUM repository or an RPM package, open the /etc/grafana/grafana.ini file.
- If Grafana is installed by using a .tar.gz file, open the {PATH_TO}/grafana-8.0.6/conf/defaults.ini file.
- ii. Find [plugins] in the configuration file to configure the allow_loading_unsigned_plugins parameter.

allow_loading_unsigned_plugins = aliyun-log-service-datasource

- 3. Restart the Grafana service.
 - i. Run the \boldsymbol{kill} command to terminate the Grafana process.
- ii. Run the following commands to start the Grafana service:
 - If Grafana is installed by using a YUM repository or an RPM package, run the following command:
 - systemctl restart grafana-server
 - If Grafana is installed by using a .tar.gz file, run the following command:
 - ./bin/grafana-server web

Step 2: Add a data source for Grafana

- The following procedure describes how to add the Log Service plug-in as a data source for Grafana.
- 1. Log on to Grafana.
- 2. In the left-side navigation pane, choose Setting > Data Sources.
- 3. On the Data Sources tab, click Add data source.
- 4. On the Add data source page, click Select in the LogService card.
- Configure the data source. The following table describes the parameters.

Parameter	Description
Name	The name of the data source.
Default	The Default switch. In this example, turn on the switch.
Endpoint	The endpoint of the Log Service project. Enter an endpoint based on your business requirements. For more information, see the Obtain the endpoint of Log Service topic in Preparations .
Project	The name of the project.
Logstore	The name of the Logstore.
AccessKeyId	The AccessKey ID provided by Alibaba Cloud. The AccessKey ID is used to identify the user. To ensure the security of your account, we recommend that you use the AccessKey pair of a RAM user. For more information, see the Obtain AccessKey credentials topic in Preparations .
AccessKeySecret	The AccessKey secret provided by Alibaba Cloud. The AccessKey secret is used to authenticate the key of the user. To ensure the security of your account, we recommend that you use the AccessKey pair of a RAM user. For more information, see the Obtain AccessKey credentials topic in Preparations .

6. Click Save & Test.

Step 3: View imported trace data

- The following procedure describes how to view the trace data imported from Log Service:
- 1. In the left-side navigation pane, choose Explore.
- 2. In the upper-left corner of the **Explore** page, select the data source.
- 3. Enter trace in the xcol(time) field. Then, click **Run query** in the upper-right corner.

							🛄 Split		hour ~ Q	🛱 Clear all 式	🔾 Run query 🗸
a302804310	c94e2774c68959e0bdfce6b										
ycol			xcol(time)	trace							
+ Add que	ery 🔊 Query history 🕥 I	Inspector									
Trace View											
front-end	: POST /orders a30280431c94e27	74c68959e0bdfce6b									
	HE 20 2021 00.27:00 880 Dure	nian 22 dama - Candada B. Danthi	Tatal Casas 52								
Oms	-CH 20 2021, 09.37.00.889 Dula	8.11ms	rotar spans 33								
				_							
Service &	Operation	✓ > < > 0ms			8.11ms		16.22ms	±			32.44ms
v front-en	nd POST /orders										
√ front-en fron	nd POST /orders nt-end request handler - /orders	I 0.01ms									
✓ front-en fron ✓ fron ✓ fron	nd POST/orders It-end request handler - /orders It-end HTTP GET	I 0.01ms	3.5	52ms							
✓ front-en fron ✓ fron ✓ fron	nd POST /orders It-end request handler - /orders It-end HTTP GET	I 0.01ms	GET	52ms				Servic	e: front-end Di	uration: 3.52ms Sta	art Time: 1.19ms
✓ front-en fron ✓ fron	nd POST/orders nt-end request handler - /orders nt-end HTTP GET	I 0.01ms	GET	32ms				Servic	e: front-end D	uration: 3.52ms Sta	art Time: 1.19ms
<pre>> front-en fron > fron </pre>	nd POST/orders itend request handler-/orders itend HTTP GET	I DOTINS HTTP > Tags:	GET stp.flavor = 1.1 http.	32ms host=user:80 ht	ttp.method - GET http.response	content_length_unc	compressed = 386 http status_t	Servic code = 200 http.st	e: front-end Dr atus_text = OK	uration: 3.52ms Sta http.target = /custo	art Time: 1.19ms mers/57a98d9_
 front-er fron fron 	nd POST/orders Ittend request handler - /orders Ittend HTTP GET	1 DOTINS HTTP > Tags: > Process	GET http:flavor=1.1 http: i: telemetry.sdk.langu	32ms host = user:80 ht rage = nodejs tele	ttp.method = GET http.response emetry.sdk.name = opentelemetry	t_content_length_unc	compressed = 386 http.status_t sion = 0.18.2 host =	Servic code = 200 http.st	e: front-end Di atus_text = OK	uration: 3.52ms Sta http.target = /custo Sean10-5	art Time: 1.19ms mers/57a98d9
<pre>> front-er fron > fron > fron </pre>	nd POST /orders Htend request handler - /orders ttend HTTP GET USEF GET /outsomers	Looins HTTP > Tage: > Proces	GET http:flavor = 1.1 http: s: telemetry.sdk.langu 1.08ms	š2ms host = user:80 ht iage = nodejs ∶ tele	ttp.method ⊂ GET = http.response emetry.sdk.name = opentelemetry	e_content_length_unc	compressed = 386 http.status_r sion = 0.18.2 host =	Servic	e: front-end Di atus_text = OK	uration: 3.52ms Sta http.target = /custo SpaniD: 5	art Time: 1.19ms mers/57a98d9_ 192a433b504bd5e4
v front-en	nd POET/ordens Itemd regeat handler / ordens Itemd HTTP GET USEF (CET/costomers USEF (CET/costomers USEF (ctuest)	I DOTING HTTP > Tags: > Proces	a.e GET http:flavor = 1.1 http: s: telemetry.sdk.langu 1.08ms 1.07ms	32ms host = user:80 h1 hage = nodejs tele	ttp.method = GET http.response emetry.sdk.name = opentelemetry	r_content_length_und	compressed = 386 [–] http status, i sion = 0.18.2 [–] host =	Servic	e: front-end Di	uration: 3.52ms St. http.target = /custo Spanit: 5	art Time: 1.19ms mers/57a98d9 192a433b504bd5e4
<pre> front-en fron fron fron</pre>	nd PoET /ordens fand regest handler /ordens handler /ITP OET user OET /outforms / user get users user of users user of users	1 adims HTTP 2 Tags: 2 Proces	3.5 GET http:flavor = 1.1 http: 1: telemetry sdk.langu 1: 08ms 1: 07ms 0: 0.3ms	12ms host = user:80 ht hage = nodejs tele	ttp.method = GET http:response emetry.sdk.name = opentelemetry	r_content_length_unc	compressed - 386 http:status_ sion = 0.18.2 host =	Servic	e: front-end Di	uration; 3.52ms Stu http.target = /custo spanit: 5	art Time: 1.19ms mers/57a98d9 192a433b504bd5e4
<pre>> fronter fron</pre>	nd POET /orders t-tend regest handler /orders teed wiTP DET user GET /outdomens (User and form the user form the u	L doins HTTP > Tage: > Proces	3.5 GET http:flavor = 1.1 http: i: telemetry.sdk.langu 1.08ms 1.07ms 0.88ms 0.07ms	52ms host = user:80 ht lage = nodejs tele	ttp.method = GET http:response emetry:sdk.name -opentelemetry	_content_length_unc	compressed = 386 http.status_i	Servic	e: front-end Dr	uration; 3.52ms Stu http.target = /custo spent0: 5	art Time: 1.19ms mers/57a98d9 192a433b504bd5e4
<pre>v fronter fron fron v fro</pre>	nd POET /ordens Intend negast handler /ordens Intend INTEP GET USEF OET /contouring	L D.D.Ims HTTP > Tags: > Proces	GET ttp:flavor = 1.1 http. ttp:flavor = 1.1 http. 1.08ms 1.07ms 0.03ms 0.03ms 0.03ms	32ms host = user:80 ht rage = nodejs tele	ttp.method = GET http.response emetry sdk.name - opentelemetry # 2.87ms	content_length_uno	compressed = 386 http statur_, alon = 0.18.2 hest =	Servic	atus_text = OK	uration: 3.52ms Sti http.target = /custo spenit 5	art Time: 1.19ms mers/57a98d9_ 192a433b504bd5e4
<pre>v fronter</pre>	nd POET/ordens Itemd regest handler /ordens itemd HTTP GET Usef of Users Users Users Usef of Users U	1 adim HTTP > Tags: > Proces	S.S. GET http:flavor = 1.1 http: i: telemetry.sck.langu 1.08ms 1.07ms 0.08ms 0.08ms 0.08ms	s2ms host = user:80 ht isage = nodejs tele	ttp.method = GET http:response emetry.sdk.name - opendemetry 2 .67ms	content_length_unc	compressed - 386 http status, j sion = 0.182 host =	Servic	atus_text = OK	uration: 3.52ms Sti http.target = /custo sponto: 5	art Time: 1.19ms mers/57a98d9 192a433b504bd5e4

4.9.1.8.2. Import trace data from Apache SkyWalking to Log Service based on an

ACK cluster

This topic describes how to import trace data from Apache SkyWalking by using a Logtail that is deployed in an Alibaba Cloud Container Service for Kubernetes (ACK) cluster to Log Service. After the trace data is collected to Log Service, you can store, analyze, visualize the data in Log Service. You can also perform AlOps based on the data.

Prerequisites

- A Logstore is created. For more information, see Create a Logstore.
- A machine group is created, and the group uses a custom identifier. For more information, see Create a custom identifier-based machine group.
 - ① Important Make sure that the custom identifier is unique in the region of the Log Service project to which the Logstore belongs.
- A trace instance is created. For more information, see Create a trace instance.

Step 1: Deploy a data collection image

 Deploy the Logtail image to an ACK cluster by using a configuration file. The following sample code provides a configuration template.

containers:	
- name: logtail	
<pre># more info: https://cr.console.a</pre>	liyun.com/repository/cn-hangzhou/log-service/logtail/detail
image: registry.cn-hangzhou.aliyu	ncs.com/log-service/logtail:v0.16.68.0-7a79f4e-aliyun
command:	
- sh	
c	
- /usr/iocal/iiogtaii/run_iogtaii	.sn 10
iivenessriobe:	
exec.	
- (sta/ipit_d/ilogtaild	
= etatue	
initialDelaySeconds: 30	
periodSeconds: 30	
resources:	
limits:	
memory: 512Mi	
requests:	
cpu: 10m	
memory: 30Mi	
env:	
- name: "ALIYUN_LOGTAIL_USER_ID	,"
value: "\${your_aliyun_user_id	}"
- name: "ALIYUN_LOGTAIL_USER_DE	FINED_ID"
value: "\${your_machine_group_	user_defined_id} "
- name: "ALIYUN_LOGTAIL_CONFIG"	
value: "/usr/local/ilogtail/i	logtail_config.json"
- name: "ALIYUN_LOG_ENV_TAGS"	
value: "_pod_name_ _pod_ip_ _	namespace_ _node_name_ _node_ip_"
- name: "_pod_name_"	
valueFrom:	
fieldRef:	
fieldPath: metadata.name	
- name: "_pod_ip_"	
valueFrom:	
fieldRef:	
fieldPath: status.podiP	
- name:namespace	
fieldRef.	
fieldPath: metadata names	Dace
- name: " node name "	pace
valueFrom:	
fieldRef:	
fieldPath: spec.nodeName	
- name: " node ip "	
valueFrom:	
fieldRef:	
fieldPath: status.hostIP	
arameter	Description
former alignment into	Enter the ID of your Alibaba Cloud account.
{your_allyun_user_la}	You can view the ID in the /etc/ilogtail/users directory.
	Enter the custom identifier that you specify when you create the machine group.

2. Redeploy the container where Logtail runs.

\${your_machine_group_user_defined_id}

Step 2: Create a Logtail configuration

the Logstore belongs.

() Important Make sure that the custom identifier is unique in the region of the Log Service project to which

Ρ

\$

1. Log on to the Log Service console.

- 2. In the Import Data section, click SkyWalking.
- 3. In the Specify Logstore step, select the project and Logstore. Then, click Next.
- 4. In the Create Machine Group step, click Use Existing Machine Groups.
- In the Machine Group Settings step, select the machine group that you want to use in the Source Server Groups section and move the machine group to the Applied Server Groups section. Then, click Next.
- 6. In the Specify Data Source step, configure the Logtail plug-ins and click Next.
- A configuration template is provided in the **Plug-in Config** field. You must replace *\${instance}* with the ID of your trace instance.

```
"inputs" : [
   {
        "detail" : {
            "Address" : "0.0.0.0:11800"
        }.
        "type" : "service_skywalking_agent_v3"
   }
],
"aggregators" : [
        "detail" : {
            "MetricsLogstore" : "${instance}-metrics",
            "TraceLogstore" : "${instance}-traces"
        }.
        "type" : "aggregator skywalking"
   }
"global" : {
   "AlwaysOnline" : true,
    "DelayStopSec" : 300
```

Click Next to complete the Logtail configuration. Then, Log Service starts to collect logs.

(?) Note A Logtail configuration requires up to 3 minutes to take effect.

7. Preview data, configure indexes, and then click Next.

By default, full-text indexing is enabled for Log Service. You can also configure field indexes based on collected logs in manual mode or automatic mode. To configure field indexes in automatic mode, click **Automatic Index Generation**. This way, Log Service automatically creates field indexes. For more information, see Configure indexes.

() Important If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.

 Click Log Query. You are redirected to the query and analysis page of your Logstore. You must wait approximately 1 minute for the indexes to take effect. Then, you can view the collected logs on the Raw Logs tab. For more information, see Query and analyze logs.

FAQ

How do I check whether a Logtail configuration is in effect?

- $\label{eq:local_$
- If the command output includes **SkyWalking**, the Logtail configuration is in effect.
- If the command output does not include **SkyWalking**, the Logtail configuration is not in effect. In this case, check whether the custom identifier that you specify for the machine group when you deploy the data collection image is the same as the custom identifier that you specify when you create the machine group.

What to do next

After you complete the preceding configurations, you can query and analyze the trace data of Apache SkyWalking in the specified Logstore. For more information, see Query and analyze trace data.

4.9.1.8.3. Import Ingress trace data from Kubernetes clusters to Log Service

This topic describes how to import Ingress trace data from Kubernetes clusters to the Trace application of Log Service by using OpenTelemetry.

Prerequisites

A trace instance is created. For more information, see Create a trace instance.

Step 1: Install the OpenTelemetry Collector

- 1. Log on to your Kubernetes cluster.
- 2. Install cert-manager.

kubectl apply -f https://github.com/jetstack/cert-manager/releases/download/v1.6.1/cert-manager.yaml

- 3. Deploy the OpenTelemetry Operator.
 - i. Download the opentelemetry-operator.yaml file.

wget https://github.com/open-telemetry/opentelemetry-operator/releases/latest/download/opentelemetry-operator.yaml

ii. Open the opentelemetry-operator.yaml file and replace the image information in the file. Replace ghcr.io/open-telemetry/opentelemetry-operator/opentelemetry-operator in the opentelemetry-operator.yaml file with sls-registry.cn-beijing.cr.aliyuncs.com/opentelemetry-operator/opentelemetry-operator . iii. Run the following command to apply the configuration:

kubectl apply -f opentelemetry-operator.yaml

4. Deploy the OpenTelemetry Collector.

i. Create a YAML file.

vim collector.yaml

a k m

ii. Enter the following code in the YAML file and configure the parameters based on your business scenario:

niVersion: opentelemetry io/vlalphal
ind: OpenTelemetryCollector
etadata.
name: otel
name. occi
image: otel/opentelemetry=collector=contrib:latest
config:
receivers.
otlp:
protocols:
groc.
gipe.
iseger.
protocole.
proceeds.
dipe.
thrift compact.
thrift biparu.
chilic_binary.
exporters.
alibabacloud logservice/logs.
endpoint: "cn=bangzhou log aliwuncs com"
project: "demo=project"
logstore: "store=logs"
access key id. "access=key=id"
access_key_id. access key id
alibabacloud logservice/metrics:
endpoint: "cn=bangzhou log aliwungs com"
project: "demo=project"
logstore: "store=traces=metrics"
access key id. "access=key=id"
access_key_id. access key id
alibabacloud loggervice/traces.
endpoint: "cn=bangzhou log alivuncs com"
project: "demo-project"
logstore: "store=traces"
access key id. "access-key-id"
access key secret: "access=key=secret"
accept_rej_cectet. accept rej sectet
service:
pipelines:
traces:
receivers: [otlp, jaeger, zipkin]
exporters: [alibabacloud logservice/traces]
metrics:
receivers: [otlp]
exporters: [alibabacloud logservice/metrics]

Parameter	Description	
endpoint	The Log Service endpoint. For more information, see the Obtain the endpoint of Log Service topic in Preparations .	
project	The name of the project that you specify when you create a trace instance. For more information, seeCreate a trace instance.	
logstore	The name of the Logstore. After you create a trace instance, Log Service generates three Logstores in the specified project to store log data, metric data, and trace data. Specify the Logstore name based on your business requirements. <i>trace_instance_id</i> -logs <i>trace_instance_id</i> -traces-metrics <i>trace_instance_id</i> -traces <i>trace_instance_id</i> -specifies the ID of the trace instance. For more information, seeCreate a trace instance.	
access_key_id	 The AccessKey ID that is used to access Log Service. We recommend that you use the AccessKey pair of a RAM user that has only the write permissions on the Log Service project. An AccessKey pair consists of an AccessKey ID and an AccessKey secret. For information about how to grant a RAM user the write permissions on the Log Service project, seRAM management. For more information, see the Obtain AccessKey credentials topic in Preparations. 	
access_key_secret	The AccessKey secret that is used to access Log Service.	

Run the following command to apply the configuration.
 otel-test indicates the namespace where your service resides.

kubectl apply -f collector.yaml --namespace=otel-test

Step 2: Configure Ingress OpenTracing

In this example, an Alibaba Cloud Container Service for Kubernetes (ACK) cluster is used.

- 1. Log on to the ACK console.
- For more information, see Log on to the ACK console in the User Guide for Container Service for Kubernetes.
- 2. On the **Clusters** page, click the cluster that you want to manage.
- 3. In the left-side navigation pane, choose Configurations > ConfigMaps.
- 4. On the **ConfigMap** page, select **kube-system** from the Namespace drop-down list. In the ConfigMap list, find **nginx-configuration** and click **Edit** in the Actions column.
- 5. In the **Edit** panel, configure the following two parameters and click **OK**.

otel-test indicates the namespace where your service resides. This namespace must be the same as the namespace that you specified in Step 1.

zipkin-collector-host: otel-collector.otel-test.svc.cluster.local:9411/api/v1/spans? enable-opentracing: true

After you complete the preceding configuration, OpenTelemetry uploads the Ingress trace data that is generated by your Kubernetes cluster to Trace. You can view the trace data in the Trace application. For more information, see View trace details.

4.9.1.9. FAQ

4.9.1.9.1. How do I implement OpenTelemetry automatic instrumentation in a

Kubernetes cluster?

This topic describes how to implement OpenTelemetry automatic instrumentation in a Kubernetes cluster to upload trace data to Log Service.

Procedure

- 1. Log on to your Kubernetes cluster.
- 2. Run the following command to install cert-manager:

kubectl apply -f https://github.com/cert-manager/cert-manager/releases/download/v1.9.1/cert-manager.yaml

3. Run the following command to install OpenTelemetry Operator:

kubectl apply -f https://github.com/open-telemetry/opentelemetry-operator/releases/latest/download/opentelemetry-operator.yaml

The opentelemetry-operator.yaml configuration file specifies image addresses at ghcr.com. If you visit the addresses in China, slow access or errors may occur. To accelerate image access, you can change the addresses in the opentelemetry-operator.yaml configuration file.

4. Run the following command to install OpenTelemetry Collector:

kubectl apply -f - < <eof< td=""></eof<>
apiVersion: opentelemetry.io/v1alpha1
kind: OpenTelemetryCollector
metadata:
name: otel
spec:
<pre>image: otel/opentelemetry-collector-contrib:latest</pre>
config:
receivers:
otlp:
protocols:
grpc:
http:
exporters:
alibabacloud_logservice/logs:
endpoint: "cn-hangzhou.log.aliyuncs.com"
project: "demo-project"
logstore: "store-logs"
access_key_id: "access-key-id"
access_key_secret: "access-key-secret"
alibabacloud_logservice/metrics:
endpoint: "cn-hangzhou.log.aliyuncs.com"
project: "demo-project"
logstore: "store-traces-metrics"
access_key_id: "access-key-id"
access_key_secret: "access-key-secret"
alibabacloud_logservice/traces:
endpoint: "cn-hangzhou.log.aliyuncs.com"
project: "demo-project"
logstore: "store-traces"
access_key_id: "access-key-id"
access_key_secret: "access-key-secret"
service.
ninelines.
traces.
receivers: [ot]n]
exporters: [alibabacloud logservice/traces]
logs:
receivers: [otlp]
exporters: [alibabacloud logservice/logs]
metrics:
receivers: [otlp]
exporters: [alibabacloud logservice/metrics]
EOF

The following table describes the parameters in the command. You can modify the parameters based on your business requirements.

Parameter	Description	
endpoint	The Log Service endpoint. For more information, see the Obtain the endpoint of Log Service topic in Preparations .	
project	The name of the project that you specify when you create a trace instance. For more information, seeCreate a trace instance.	
logstore	The name of the Logstore. After you create a trace instance, Log Service generates three Logstores in the specified project to store log data, metric data, and trace data. Specify the Logstore name based on your business requirements. • trace_instance_id-logs • trace_instance_id-traces-metrics • trace_instance_id-traces trace_instance_id-traces trace_instance_id-traces	
access_key_id	The AccessKey ID that is used to access Log Service. For more information, see the Obtain AccessKey credentials topic in Preparations .	
access_key_secret	The AccessKey secret that is used to access Log Service. For more information, see the Obtain AccessKey credentials topic in Preparations .	

5. Run the following command to install OpenTelemetry Auto-Instrumentation:

kubectl apply -f - <<EOF apiVersion: opentelemetry.io/v1alpha1 kind: Instrumentation metadata: name: my-java-instrumentation spec: exporter: endpoint: http://otel-collector:4317 propagators: tracecontext - baggage - b3 java: image: ghcr.io/open-telemetry/opentelemetry-operator/autoinstrumentation-java:latest nodejs: image: ghcr.io/open-telemetry/opentelemetry-operator/autoinstrumentation-nodejs:latest python: image: ghcr.io/open-telemetry/opentelemetry-operator/autoinstrumentation-python:latest dotnet: image: ghcr.io/open-telemetry/opentelemetry-operator/autoinstrumentation-dotnet:latest env: - name: OTEL RESOURCE ATTRIBUTES value: service.name=your_service,service.namespace=your_service_namespace EOF

 You can use the Auto-Instrumentation environment variable to pass parameters such as the name and namespace of your service to the command. If multiple applications exist, we recommend that you create an Auto-Instrumentation for each application to differentiate service names.

(?) Note The value of the metadata.name parameter for each Auto-Instrumentation must be unique.

- For applications in different programming languages such as Java, Node.js, Python, and .NET, configure Auto-Instrumentation based on your business requirements.
- 6. Add the configuration for automatic instrumentation to your configuration file.
 - Add the configuration script to the configuration file of your application based on your business requirements. Only Python, Node.js, Java, and .NET applications are supported.

() **Important** In the following code, <u>my-java-instrumentation</u> is the name of the Auto-Instrumentation that you install in the preceding step. The Auto-Instrumentation name is the value of the **metadata.name** parameter. You can change the name based on your business requirements.

type: Kollingupaate

nnotations:	
instrument	ation.opentelemetry.io/inject-java:"my-java-instrumentation
instrument	ation.opentelemetry.io/inject-sdk:"my-java-instrumentation"
abels:	
app: tea	
ec:	
containers:	

∘ Java

instrumentation.opentelemetry.io/inject-java: "my-java-instrumentation"
instrumentation.opentelemetry.io/inject-sdk: "my-java-instrumentation"

Python

instrumentation.opentelemetry.io/inject-python: "my-java-instrumentation"
instrumentation.opentelemetry.io/inject-sdk: "my-java-instrumentation"

• Node.js

instrumentation.opentelemetry.io/inject-nodejs: "my-java-instrumentation"
instrumentation.opentelemetry.io/inject-sdk: "my-java-instrumentation"

dotNET

instrumentation.opentelemetry.io/inject-dotnet: "my-java-instrumentation"
instrumentation.opentelemetry.io/inject-sdk: "my-java-instrumentation"

4.10. Time series storage

4.10.1. Data import

4.10.1.1. Collect metric data from hosts

Log Service allows you to collect metric data from hosts by using Logtail. The metric data includes CPU, memory, load, disk, and network data. This topic describes how to create a Logtail configuration in the Log Service console to collect metric data from hosts.

Prerequisites

Logtail V0.16.40 or later is installed on a Linux server. For more information, see Install Logtail on a Linux server.

Limits

- Windows servers are not supported.
- Metric data related to GPUs and hardware status cannot be collected.

Procedure

- 1. Log on to the Log Service console.
- 2. In the Import Data section, click Host Monitoring Data.
- 3. In the Specify Logstore step, select the project and the Metricstore that you want to use. Then, click Next.
- You can also click Create Now to create a project or a Metricstore. For more information, see Create a project and Create a Metricstore.
- 4. In the **Create Machine Group** step, create a machine group.
 - If a machine group is available, click Use Existing Machine Groups.
 - If no machine groups are available, perform the following steps to create a machine group. In this example, an Elastic Compute Service (ECS) instance is used.
 - a. Select the ECS instance on which you want to install Logtail. For more information, see Install Logtail on a Linux server. If Logtail is installed on the ECS instance, click **Complete Installation**.

O Note If you want to collect logs from self-managed clusters or servers that are provided by third-party cloud service providers, you must manually install Logtail. For more information, see Install Logtail on a Linux server.

- b. After Logtail is installed, click Complete Installation.
- c. Create a machine group.

For more information about how to create a machine group, see Create an IP address-based machine group or Create a custom identifier-based machine group.

5. Select the machine group from the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

① Important If you enable a machine group immediately after you create the machine group, the heartbeat status of the machine group may be FAIL. This issue occurs because the machine group is not connected to Log Service. To resolve this issue, you can click Automatic Retry. If the issue persists, troubleshoot the issue. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail Client?.

6. In the Specify Data Source step, configure the Config Name and Plug-in Config parameters. inputs: specifies the collection configurations. This parameter is required. Configure the inputs parameter based on your data source.

Note You can specify only one type of data source in the **inputs** parameter.

Parameter	Туре	Required	Description
type	string	Yes	The type of the data source. Set the value to metric_system_v2.
IntervalMs	int	Yes	The interval between two consecutive requests. Unit: milliseconds. The value must be greater than or equal to 5000. We recommend that you set the value to 30000.

7. Click Next

Metrics

The following tables describe metrics that are related to CPUs, memory, loads, disks, and networks.

CPU-related metrics

Metric	Description	Unit	Example
cpu_count	The number of CPU cores.	N/A	2.0
cpu_util	The CPU utilization. The CPU utilization is equal to one minus the sum of the idle, wait, and steal counters.	Percent (%)	7.68
cpu_guest_util	The guest counter of Linux. This counter indicates the percentage of the time that is spent by the CPU on processes with normal priority.	Percent (%)	0.0
cpu_guestnice_util	The guest_nice counter of Linux. This counter indicates the percentage of the time that is spent by the CPU on processes with nice priority.	Percent (%)	0.0
cpu_irq_util	The irq counter of Linux. This counter indicates the percentage of the time that is spent by the CPU to serve hardware interrupt requests.	Percent (%)	0.0

cpu_nice_util	The nice counter of Linux. This counter indicates the percentage of the time that is spent by the CPU on user-mode processes with nice priority.	Percent (%)	0.0
cpu_softirq_util	The softirq counter of Linux. This counter indicates the percentage of the time that is spent by the CPU to serve software interrupt requests.	Percent (%)	0.06
cpu_steal_util	The steal counter of Linux. This counter indicates the percentage of the time that is spent by the CPU to run other operating systems in a virtual environment.	Percent (%)	0.0
cpu_sys_util	The system counter of Linux. This counter indicates the percentage of the time that the CPU spends on kernel-mode processes.	Percent (%)	2.77
cpu_user_util	The user counter of Linux. This counter indicates the percentage of the time that is spent by the CPU on user-mode processes with normal priority.	Percent (%)	4.84
cpu_wait_util	The iowait counter of Linux. This counter indicates the percentage of the CPU idle time when outstanding disk I/O requests exist.	Percent (%)	0.11

Memory-related metrics

Metric	Description	Unit	Example
mem_util	The memory usage.	Percent (%)	51.03
mem_cache	The amount of memory that is allocated but unused.	byte	3566386668.0
mem_free	The amount of the unused memory.	byte	177350084.0
mem_available	The amount of the available memory.	byte	3699885553.0
mem_used	The amount of the used memory.	byte	4041510463.0
mem_swap_util	The swap memory usage.	Percent (%)	0.0
mem_total	The size of the memory.	byte	7919128576.0

Disk-related metrics

Metric	Description	Unit	Example
disk_rbps	The amount of data that is read from the disk per second.	byte/s	8376.81
disk_wbps	The amount of data that is written to the disk per second.	byte/s	247633.58
disk_riops	The number of read operations that are completed on the disk per second.	N/A	0.22
disk_wiops	The number of write operations that are completed on the disk per second.	N/A	43.39
disk_rlatency	The average read latency.	ms	2.83
disk_wlatency	The average write latency.	ms	2.15
disk_util	The I/O usage of the disk.	Percent (%)	0.27
disk_space_usage	The percentage of the used disk space.	Percent (%)	9.12
disk_inode_usage	The percentage of the used index node (inode) space.	Percent (%)	1.18
disk_space_used	The amount of the used disk space.	byte	11068512238.59
disk_space_total	The total amount of the disk space.	byte	126692061184.0
disk_inode_total	The total amount of the inode space.	byte	7864320.0
disk_inode_used	The amount of the used inode space.	byte	93054.78

Network-related metrics

Metric	Description	Unit	Example
net_drop_util	The percentage of the number of discarded packets to the total number of packets.	Percent (%)	0.0
net_err_util	The percentage of the number of error packets to the total number of packets.	Percent (%)	0.0
net_in	The amount of data that is received per second.	byte/s	8440.91

User Guide-Log Service

net_in_pkt	The number of packets that are received per second.	N/A	40.83
net_out	The amount of data that is sent per second.	byte/s	12446.53
net_out_pkt	The number of packets that are sent per second.	N/A	39.95

TCP-related metrics

Metric	Description	Unit	Example
protocol_tcp_established	The number of established connections.	N/A	205.0
protocol_tcp_insegs	The number of received packets.	N/A	4654.0
protocol_tcp_outsegs	The number of sent packets.	N/A	4870.0
protocol_tcp_retran_segs	The number of re-sent packets.	N/A	0.0
protocol_tcp_retran_util	The percentage of the number of re-sent packets to the number of sent packets.	Percent (%)	0.0

• System-related metrics

Metric	Description	Unit	Example
system_boot_time	The startup time of the system.	s	1578461935.0
system_load1	The average system load every minute.	N/A	0.58
system_load5	The average system load every 5 minutes.	N/A	0.68
system_load15	The average system load every 15 minutes.	N/A	0.60

What to do next

- Query and analyze data
- After metric data is collected, you can query and analyze the data on the Query & Analysis page of the Metricstore. For more information, see Query and analyze time series data.
- · Visualize query and analysis results in Log Service

You can create a time series chart and add the chart to a dashboard to visualize the query and analysis results. You can also configure alert rules for the chart.

4.10.1.2. Collect ping and tcping data

This topic describes how to use Logtail to collect ping and toping data to a Metricstore of Log Service.

Prerequisites

A project and a Metricstore are created. For more information, see Create a project and Create a Metricstore.

Limits

Only Linux Logtail V1.0.31 and later can collect ping and tcping data. If you installed an earlier version of Logtail on your server, you must update Logtail first.

Procedure

- 1. Log on to the Log Service console.
- 2. In the Projects section, click the project that you want to manage.
- 3. In the left navigation sidebar, click **Time Series Storage**.
- 4. Find the Metricstore and choose Data Import > Logtail Configurations. Then, click the plus sign (+) to the right of Logtail Configurations.
- 5. In the Import Data dialog box, click Custom Data Plug-in.
- You can also click View More and enter Custom Data Plug-in in the search field of Import Data.
- 6. In the Create Machine Group step, create a machine group.
- If a machine group is available, click Use Existing Machine Groups.
- If no machine groups are available, perform the following steps to create a machine group. In this example, an Elastic Compute Service (ECS) instance is used.
- a. Select the ECS instance on which you want to install Logtail. For more information, see Install Logtail on a Linux server. If Logtail is installed on the ECS instance, click Complete Installation.

② Note If you want to collect logs from self-managed clusters or servers that are provided by third-party cloud service providers, you must manually install Logtail. For more information, see Install Logtail on a Linux server.

- b. After Logtail is installed, click Complete Installation.
- c. Create a machine group

For more information about how to create a machine group, see Create an IP address-based machine group or Create a custom identifier-based machine group.

7. Select the machine group from the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

① Important If you enable a machine group immediately after you create the machine group, the heartbeat status of the machine group may be FAIL. This issue occurs because the machine group is not connected to Log Service. To resolve this issue, you can click Automatic Retry. If the issue persists, troubleshoot the issue. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail client?.

 In the Specify Data Source step, configure the Config Name and Plug-in Config parameters. Then, click Next. inputs is required and is used to configure the data source settings for the Logtail configuration.

```
() Important You can specify only one type of data source in inputs.
  {
                         "inputs": [
                                                {
                                                                    "detail": {
                                                                                               "tcp": [
                                                                                                      tcp": {
    {
        "port": 80,
        "src": "192.XX.XX.103",
        "count": 3,
        "target": "www.aliyun.com"
        "target": "www.aliyun.com"

                                                                                                ],
                                                                                               "interval_seconds": 60,
"icmp": [
{
"src": "192.XX.XX.103",
                                                                                                                                            "count": 3,
"target": "www.aliyun.com"
                                                                                                                }
                                                                                           ]
                                                                       },
"type": "metric_input_netping"
                                      }
               ]
}
```

Parameter Type	Required	Description	
tcp array	Yes	<pre>Specifies to collect TCP ping data. The following fields are specific to the tcp parameter. You can configure the following fields based on your business requirements: • port: the port number. • src: the IP address of the machine on which the ping command is run. The src field specifies the machine on which the ping command is run in your machine group. • count: the number of packets that can be sent by the ping command. Valid values: (0,10). Default value: 3. We recommend that you use the default value. • target: the hostname or IP address of the machine that is pinged. The target field specifies the hostname or IP address of the destination machine. You can add multiple IP addresses. Example: "tcp": { { "port": 80, "src": "192.XX.XX.103", "count": 3, "target": "www.aliyun.com" }, { "count": 3, "target": "www.aliyun.com" }] </pre>	
icmp	array	Yes	<pre>Specifies to collect Internet Control Message Protocol (ICMP) ping data. The following fields are specific to the icmp parameter. You can configure the following fields based on your business requirements: • src: the IP address of the machine on which the ping command is run. The src field specifies the machine on which the ping command is run in your machine group. • count: the number of packets that can be sent by the ping command. Valid values: (0,10). Default value: 3. We recommend that you use the default value. • target the hostname or IP address of the machine that is pinged. The target field specifies the hostname or IP address of the destination machine. You can add multiple IP addresses. Example: "icmp": [</pre>
------------------	--------	-----	--
interval_seconds	int	Yes	The interval at which the ping command is run. Unit: seconds. • Default value: 60. • Valid values: [10,86400)
type	string	Yes	The type of the data source. Set the value to metric_input_netping.

Preview data, configure indexes, and then click Next.
 By default, full-text indexing is enabled for Log Service. You can also configure field indexes based on collected logs in manual mode or automatic mode. To configure field indexes in automatic mode, click Automatic Index Generation. This way, Log Service automatically creates field indexes. For more information, see Create indexes.

() Important If you want to query and analyze logs, you must enable full-text indexing or field indexing. If you enable both full-text indexing and field indexing, the system uses only field indexes.

What to do next

After the ping data is collected, you can query and analyze the data in the Metricstore. For more information, see Query and analyze time series data. The following table describes the metrics in the collected data.

Category	Metric	Description
	ping_failed	The number of the packets that fail to be sent when an ICMP ping command is run.
	ping_rtt_avg_ms	The average response time of the packets that are sent when an ICMP ping command is run. Unit: milliseconds.
	ping_rtt_max_ms	The maximum response time of the packets that are sent when an ICMP ping command is run. Unit: milliseconds.
ICMP ping	ping_rtt_min_ms	The minimum response time of the packets that are sent when an ICMP ping command is run. Unit: milliseconds.
	ping_rtt_stddev_ms	The standard deviation in the response time of the packets that are sent when an ICMP ping command is run. Unit: milliseconds.
	ping_rtt_total_ms	The total response time of the packets that are sent when an ICMP ping command is run. Unit: milliseconds.
	ping_success	The number of the packets that are successfully sent when an ICMP ping command is run.
	ping_total	The total number of the packets that are sent when an ICMP ping command is run.
	tcping_failed	The number of the packets that fail to be sent when a TCP ping command is run.
	tcping_rtt_avg_ms	The average response time of the packets that are sent when a TCP ping command is run. Unit: milliseconds.
	tcping_rtt_max_ms	The maximum response time of the packets that are sent when a TCP ping command is run. Unit: milliseconds.
TCP ping	tcping_rtt_min_ms	The minimum response time of the packets that are sent when a TCP ping command is run. Unit: milliseconds.
	tcpping_rtt_stddev_ms	The standard deviation in the response time of the packets that are sent when a TCP ping command is run. Unit: milliseconds.
	tcping_rtt_total_ms	The total response time of the packets that are sent when a TCP ping command is run. Unit: milliseconds.
	tcping_succcess	The number of the packets that are successfully sent when a TCP ping command is run.
	tcping_total	The total number of the packets that are sent when a TCP ping command is run.

4.10.1.3. Import metrics collected by Telegraf

4.10.1.3.1. Telegraf overview

Telegraf is an agent developed by InfluxData to collect metric data. Telegraf supports multiple input plug-ins and output plug-ins, such as MySQL, Redis, and Elasticsearch. This topic describes how Telegraf works, how to install Telegraf, and how to use Telegraf to collect metric data.

Implementation

You can use Telegraf to collect metric data from MySQL, Redis, and Elasticsearch. After you collect metric data, you can use the InfluxDB line protocol to write the collected data to Logtail. Then, Logtail uploads the metric data to a Metricstore of Log Service. Log Service allows you to configure plug-ins and create dashboards in the console for the collected data. The following figure shows how Telegraf works.

Collection methods

Telegraf provides the following collection methods:

Local collection

You can use Telegraf to collect metric data from a local server by using the local collection method. The server that is specified in your machine group is the server from which metric data is collected. When you create a Logtail configuration file, you can set the IP address of the server to 127.0.0.1. We recommend that you use this collection method.

Remote collection

You can use Telegraf that is installed on a server to collect metric data from other servers by using the remote collection method. When you create a Logtail configuration file, you can set the IP address to the actual IP address or the actual endpoint of the server. If you use the remote collection method, you can configure only one server in the machine group. If you configure more than one servers, duplicate data is generated. You can use the remote collection method in the following scenarios:

- $\circ~$ You want to collect metric data of a cloud service where you cannot install Logtail and Telegraf.
- You do not want to install a metric collection agent on a running server.

4.10.1.3.2. Collect metric data from Elasticsearch clusters

You can use Telegraf to collect metric data from Elasticsearch clusters, and then use Logtail to send the metric data to a Metricstore in Log Service. This way, you can monitor Elasticsearch in a visualized manner. This topic describes how to use Log Service to collect metric data from Elasticsearch clusters and visualize the data.

Prerequisites

- The server on which Telegraf is installed can communicate with a server in your Elasticsearch cluster over an internal network.
- A project and a Metricstore are created. For more information, see Create a project and Create a Metricstore.

Limits

Only Linux Logtail V0.16.48 and later can be used to collect metric data from Elasticsearch clusters. If you installed an earlier version of Logtail on your server, you must update Logtail to a supported version.

Procedure

- 1. Log on to the Log Service console.
- 2. In the Import Data section, click the Monitoring Data tab. Then, click Elasticsearch Monitoring Data.
- You can also click View More and enter Elasticsearch Monitoring Data in the search field of Import Data
- 3. Select the project and Metricstore and click Next.
- 4. In the Create Machine Group step, create a machine group.
 - If a machine group is available, click Use Existing Machine Groups.
 - If no machine groups are available, perform the following steps to create a machine group. In this example, an Elastic Compute Service (ECS) instance is used.
 - a. Select the ECS instance on which you want to install Logtail. For more information, see Install Logtail on a Linux server. If Logtail is installed on the ECS instance, click Complete Installation.

Note If you want to collect logs from self-managed clusters or servers that are provided by third-party cloud service providers, you must manually install Logtail. For more information, see Install Logtail on a Linux server.

b. After Logtail is installed, click Complete Installation.

c. Create a machine group.

For more information about how to create a machine group, see Create an IP address-based machine group or Create a custom identifier-based machine group.

5. Select the machine group from the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

① Important If you enable a machine group immediately after you create the machine group, the heartbeat status of the machine group may be FAIL. This issue occurs because the machine group is not connected to Log Service. To resolve this issue, you can click Automatic Retry. If the issue persists, troubleshoot the issue. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail client?.

6. In the Specify Data Source step, configure the following parameters and click Next.

Parameter	Description
Configuration Name	The name of the Logtail configuration. You can enter a custom name.
Cluster Name	The name of the Elasticsearch cluster. You can enter a custom name. After you configure this parameter, Log Service adds a <i>cluster=Cluster name</i> tag to the Elasticsearch metric data that is collected by using the Logtail configuration.
	① Important Make sure that the cluster name is unique. Otherwise, data conflicts may occur.

Server List	The information about the Elasticsearch cluster. The information includes the following configuration items:
	 Address: the address of the Elasticsearch cluster. You can enter the IP address, hostname, or domain name of the server in the cluster.
	• Port: the port number of the Elasticsearch cluster. Default value: 9200.
	You can add information about multiple Elasticsearch clusters based on your business requirements.
Password	If authentication is configured for the Elasticsearch cluster, you must enter the account and password that you use to connect to the Elasticsearch cluster.
Index Name	The name of the index that is created in the Elasticsearch cluster. If you enter_all, the metric data of all indexes in the Elasticsearch cluster is collected.
Custom Tags	The custom tags that are added to the collected Elasticsearch metric data. The tags are key-value pairs. After you configure this parameter, Log Service adds the custom tags to the Elasticsearch metric data that is collected by using the Logtail configuration.

After you complete the configuration, Log Service automatically creates assets such as Metricstores.

FAQ

How do I check whether Telegraf is collecting data as expected?

You can check the logs in the /etc/ilogtail/telegraf/telegraf.log file on your server. You can also collect the logs to Log Service and query the logs.

What to do next

Query and analysis

After the configuration is complete, Telegraf collects metric data from Elasticsearch clusters, and Logtail sends the metric data to a Metricstore in Log Service. You can query and analyze the metric data in the Metricstore. For more information, see Query and analyze time series data.

Visualization

After the configuration is complete, Log Service automatically creates a dashboard named Elasticsearch Monitoring_Cluster name in the related project. You can visualize query and analysis results on the dashboard. You can also configure alerts for the dashboard.

4.10.1.3.3. Collect metric data from MySQL servers

You can use Telegraf to collect metric data from MySQL servers, and then use Logtail to send the metric data to a Metricstore in Log Service. This way, you can monitor metric data of MySQL servers in a visualized manner. This topic describes how to collect metric data from a MySQL server by using Log Service and visualize the data

Prerequisites

- Logtail V0.16.48 or later is installed on a Linux server. For more information, see Install Logtail on a Linux server.
- Telegraf is installed on a server that is connected to the MySQL server over a private network.

Limit

Only MySQL 5.5 or later is supported.

Procedure

- 1. Log on to the Log Service console.
- 2. In the Import Data section, click MvSOL Monitoring Data.
- In the Specify Logstore step, select the project and the Metricstore that you want to use. Then, click Next. You can also click Create Now to create a project or a Metricstore. For more information, see Create a project and Create a Metricstore.
- 4. In the Create Machine Group step, create a machine group
 - If a machine group is available, click Use Existing Machine Groups.
 - If no machine groups are available, perform the following steps to create a machine group. In this example, an Elastic Compute Service (ECS) instance is used.
 - a. Select the ECS instance on which you want to install Logtail. For more information, see Install Logtail on a Linux server. If Logtail is installed on the ECS instance, click Complete Installation

O Note If you want to collect logs from self-managed clusters or servers that are provided by third-party cloud service providers, you must manually install Logtail. For more information, see Install Logtail on a Linux server.

- b. After Logtail is installed, click Complete Installation.
- c. Create a machine group

For more information about how to create a machine group, see Create an IP address-based machine group or Create a custom identifier-based

5. Select the machine group from the Source Server Groups section and move the machine group to the Applied Server Groups section. Then, click Next.

() Important If you enable a machine group immediately after you create the machine group, the heartbeat status of the machine group may be FAIL. This issue occurs because the machine group is not connected to Log Service. To resolve this issue, you can click Automatic Retry. If the issue persists, troubleshoot the issue. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail client?

6. In the Specify Data Source step, configure the parameters. The following table describes the parameters.

Parameter	Description
Configuration Name	Enter a name for the Logtail configuration.
	Enter the name of the MySQL cluster. After you configure this parameter, Log Service adds acluster= <cluster name=""> tag to the data that is collected.</cluster>
Cluster Name	⑦ Note Make sure that the cluster name is unique. Otherwise, data conflicts may occur.

Server List	Click the + icon to add the MySQL server and configure the following parameters: Account: the username of the account that is used to log on to the MySQL server. Note We recommend that you create a dedicated account to monitor the data of the MySQL server and grant the account only the permissions that are required to monitor data.
	• Password: the password of the account.
	• Address: the endpoint of the MySQL server. The endpoint can be the IP address, hostname, or domain name of the server.
	• Port: the port number of the MySQL server. The default value is 3306.
	You can add multiple MySQL servers based on your business requirements.
Custom Tags	Create a custom tag in Custom Tags . The created tag is added to the data that is collected based on the Logtail configuration. You can use custom tags to identify the data that is collected based on different Logtail configurations in a Metricstore.
	You can click the + icon to create a custom tag. You can create multiple custom tags. The custom tags are added to all data that is collected based on the Logtail configuration.

How do I check whether Telegraf is collecting data as expected?

You can check the logs of the /etc/ilogtail/telegraf/telegraf.log file on your server. You can use Log Service to collect this log file and search for the required information in Log Service.

What to do next

Query and analyze data

After you configure the settings, Telegraf uses Logtail to upload collected metric data to the specified Metricstore in Log Service. You can query and analyze the data on the Query & Analysis page of the Metricstore. For more information, see Query and analyze time series data. • Visualize guery and analysis results

You can create a time series chart and add the chart to a dashboard to view the query and analysis results. You can also configure alert rules.

4.10.1.3.4. Collect metric data from Redis databases

You can use Telegraf to collect metric data from Redis databases, and then use Logtail to send the metric data to a Metricstore in Log Service. This way, you can monitor Redis databases in a visualized manner. This topic describes how to use Log Service to collect metric data from Redis databases and visualize the data.

Prerequisites

- The server on which Telegraf is installed can communicate with your server that hosts Redis databases over an internal network.
- A project and a Metricstore are created. For more information, see Create a project and Create a Metricstore.

Limits

Only Linux Logtail V0.16.48 and later can be used to collect metric data from Redis databases. If you installed an earlier version of Logtail on your server, you must update Logtail to a supported version.

Procedure

- 1. Log on to the Log Service console.
- 2. In the Import Data section, click the Monitoring Data tab. Then, click Redis Monitoring Data.
- 3. Select the project and Metricstore and click Next.
- 4. In the Create Machine Group step, create a machine group.
 - If a machine group is available, click Use Existing Machine Groups.
 - If no machine groups are available, perform the following steps to create a machine group. In this example, an Elastic Compute Service (ECS) instance is used.
 - a. Select the ECS instance on which you want to install Logtail. For more information, see Install Logtail on a Linux server. If Logtail is installed on the ECS instance, click Complete Installation.

③ Note If you want to collect logs from self-managed clusters or servers that are provided by third-party cloud service providers, you must manually install Logtail. For more information, see Install Logtail on a Linux server.

- b. After Logtail is installed, click Complete Installation.
- c. Create a machine group

For more information about how to create a machine group, see Create an IP address-based machine group or Create a custom identifier-based machine group.

5. Select the machine group from the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

① Important If you enable a machine group immediately after you create the machine group, the heartbeat status of the machine group may be FAIL. This issue occurs because the machine group is not connected to Log Service. To resolve this issue, you can click Automatic Retry. If the issue persists, troubleshoot the issue. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail client?.

6. In the Specify Data Source step, configure the following parameters and click Next.

Parameter	Description
Configuration Name	The name of the Logtail configuration. You can enter a custom name.

Cluster Name	The name of the Redis cluster. You can enter a custom name. After you configure this parameter, Log Service adds a <i>cluster=Cluster name</i> tag to the Redis metric data that is collected by using the Logtail configuration. () Important Make sure that the cluster name is unique. Otherwise, data conflicts may occur.
Server List	 The information about the Redis database. The information includes the following configuration items: Address: the address of the Redis database. You can enter the IP address, hostname, or domain name of the server that hosts the database. Port: the port number of the Redis database. Default value: 6379. You can add information about multiple Redis databases based on your business requirements.
Password	If authentication is configured for the Redis database, you must enter the password of the Redis database.
Custom Tags	The custom tags that are added to the collected Redis metric data. The tags are key-value pairs. After you configure this parameter, Log Service adds the custom tags to the Redis metric data that is collected by using the Logtail configuration.

How do I check whether Telegraf is collecting data as expected?

You can check the logs in the /etc/ilogtail/telegraf/telegraf.log file on your server. You can also collect the logs to Log Service and query the logs.

- What to do next
- Query and analysis

After the configuration is complete, Telegraf collects metric data from Redis databases, and Logtail sends the metric data to a Metricstore in Log Service. You can query and analyze the metric data in the Metricstore. For more information, see <u>Query and analyze time series data</u>.

Visualization

After the configuration is complete, Log Service automatically creates a dashboard named *Redis Monitoring_Cluster name* in the related project. You can visualize query and analysis results on the dashboard. You can also configure alerts for the dashboard.

4.10.1.3.5. Collect metric data from MongoDB databases

You can use Telegraf to collect metric data from MongoDB databases, and then use Logtail to send the metric data to a Metricstore in Log Service. This way, you can monitor MongoDB databases in a visualized manner. This topic describes how to use Log Service to collect metric data from MongoDB databases and visualize the data.

Prerequisites

A project and a Metricstore are created. For more information, see Create a project and Create a Metricstore.

Limits

Only Linux Logtail V0.16.50 and later can be used to collect metric data from MongoDB databases. If you installed an earlier version of Logtail on your server, you must update Logtail to a supported version.

Procedure

- 1. Log on to the Log Service console.
- 2. In the Import Data section, click the Monitoring Data tab. Then, click MongoDB Monitoring.
- You can also click View More and enter MongoDB Monitoring in the search field of Import Data.
- 3. Select the project and Metricstore and click **Next**.
- 4. In the **Create Machine Group** step, create a machine group.
- If a machine group is available, click Use Existing Machine Groups
- If no machine groups are available, perform the following steps to create a machine group. In this example, an Elastic Compute Service (ECS) instance is used.
- a. Select the ECS instance on which you want to install Logtail. For more information, see Install Logtail on a Linux server. If Logtail is installed on the ECS instance, click **Complete Installation**.

③ Note If you want to collect logs from self-managed clusters or servers that are provided by third-party cloud service providers, you must manually install Logtail. For more information, see Install Logtail on a Linux server.

b. After Logtail is installed, click Complete Installation.

c. Create a machine group.

For more information about how to create a machine group, see Create an IP address-based machine group or Create a custom identifier-based machine group.

5. Select the machine group from the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

① Important If you enable a machine group immediately after you create the machine group, the heartbeat status of the machine group may be FAIL. This issue occurs because the machine group is not connected to Log Service. To resolve this issue, you can click Automatic Retry. If the issue persists, troubleshoot the issue. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail client?.

6. In the Specify Data Source step, configure the following parameters. Then, click Next.

Parameter	Description
Configuration Name	The name of the Logtail configuration. You can enter a custom name.

Cluster Name	The name of the MongoDB cluster. You can enter a custom name. After you configure this parameter, Log Service adds a <i>cluster=Cluster name</i> tag to the MongoDB metric data that is collected by using the Logtail configuration. ① Important Make sure that the cluster name is unique. Otherwise, data conflicts may occur.
Server List	 The information about the MongoDB database. The information includes the following configuration items: Address: the address of the MongoDB database. You can enter the IP address, hostname, or domain name of the server that hosts the database. Port: the port number of the MongoDB database. Default value: 3717. Account: the username of the account that you use to connect to the MongoDB database. Note We recommend that you create a dedicated account to monitor the MongoDB database and grant the account only the permissions that are required for monitoring. Password: the password of the account that you use to connect to the MongoDB database. You can add information about multiple MongoDB database based on your business requirements.
Custom Tags	The custom tags that are added to the collected MongoDB metric data. The tags are key-value pairs. After you configure this parameter, Log Service adds the custom tags to the MongoDB metric data that is collected by using the Logtail configuration.

How do I check whether Telegraf is collecting data as expected?

You can check the logs in the /etc/ilogtail/telegraf/telegraf.log file on your server. You can also collect the logs to Log Service and query the logs.

What to do next

• Query and analysis

After the configuration is complete, Telegraf collects metric data from Elasticsearch clusters, and Logtail sends the metric data to a Metricstore in Log Service. You can query and analyze the metric data in the Metricstore. For more information, see Query and analyze time series data.

Visualization

After the configuration is complete, Log Service automatically creates a dashboard named *MongoDB Monitoring_Cluster name* in the related project. You can visualize query and analysis results on the dashboard. You can also configure alerts for the dashboard.

4.10.1.3.6. Collect metric data from ClickHouse databases

You can use Telegraf to collect metric data from ClickHouse databases, and then use Logtail to send the metric data to a Metricstore in Log Service. This way, you can monitor ClickHouse databases in a visualized manner. This topic describes how to use Log Service to collect metric data from ClickHouse databases and visualize the data.

Prerequisites

- The server on which Telegraf is installed can communicate with your server that hosts ClickHouse databases over an internal network
- A project and a Metricstore are created. For more information, see Create a project and Create a Metricstore.

Limits

Only Linux Logtail V0.16.48 and later can be used to collect metric data from ClickHouse databases. If you installed an earlier version of Logtail on your server, you must update Logtail to a supported version.

Procedure

- 1. Log on to the Log Service console.
- 2. In the Import Data section, click the Monitoring Data tab. Then, click ClickHouse Monitoring Data.
- You can also click View More and enter ClickHouse Monitoring Data in the search field of Import Data.
- 3. Select the project and Metricstore and click Next.
- 4. In the **Create Machine Group** step, create a machine group.
 - If a machine group is available, click Use Existing Machine Groups.
 - If no machine groups are available, perform the following steps to create a machine group. In this example, an Elastic Compute Service (ECS) instance is used.
 - a. Select the ECS instance on which you want to install Logtail. For more information, see Install Logtail on a Linux server. If Logtail is installed on the ECS instance, click Complete Installation.

③ **Note** If you want to collect logs from self-managed clusters or servers that are provided by third-party cloud service providers, you must manually install Logtail. For more information, see Install Logtail on a Linux server.

- b. After Logtail is installed, click Complete Installation.
- c. Create a machine group

For more information about how to create a machine group, see Create an IP address-based machine group or Create a custom identifier-based machine group.

5. Select the machine group from the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

① Important If you enable a machine group immediately after you create the machine group, the heartbeat status of the machine group may be FAIL. This issue occurs because the machine group is not connected to Log Service. To resolve this issue, you can click Automatic Retry. If the issue persists, troubleshoot the issue. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail client?.

6. In the Specify Data Source step, configure the following parameters. Then, click Next.

Parameter

Description

Configuration Name	The name of the Logtail configuration. You can enter a custom name.
Cluster Name	The name of the ClickHouse cluster. You can enter a custom name. After you configure this parameter, Log Service adds a <i>cluster=Cluster name</i> tag to the ClickHouse metric data that is configuration.
	() Important Make sure that the cluster name is unique. Otherwise, data conflicts may occur.
Username	The username of the account that you use to connect to the ClickHouse database.
Password	The password of the account that you use to connect to the ClickHouse database.
Server List	 The information about the ClickHouse database. The information includes the following configuration items: Address: the address of the ClickHouse database. Port: the port number of the ClickHouse database. Default value: 8123.
	You can add information about multiple ClickHouse databases based on your business requirements.
Custom Tags	The custom tags that are added to the collected ClickHouse metric data. The tags are key-value pairs. After you configure this parameter, Log Service adds the custom tags to the ClickHouse metric data that is collected by using the Logtail configuration.

How do I check whether Telegraf is collecting data as expected?

You can check the logs in the /etc/ilogtail/telegraf/telegraf.log file on your server. You can also collect the logs to Log Service and query the logs.

What to do next

• Query and analysis

After the configuration is complete, Telegraf collects metric data from Elasticsearch clusters, and Logtail sends the metric data to a Metricstore in Log Service. You can query and analyze the metric data in the Metricstore. For more information, see Query and analyze time series data.

Visualization

After the configuration is complete, Log Service automatically creates a dashboard named *ClickHouse Monitoring_Cluster name* in the related project. You can visualize query and analysis results on the dashboard. You can also configure alerts for the dashboard.

4.10.1.3.7. Collect metric data from Kafka servers

You can use Telegraf to collect metric data from Kafka servers, and then use Logtail to send the metric data to a Metricstore in Log Service. This way, you can monitor Kafka servers in a visualized manner. This topic describes how to use Log Service to collect metric data from Kafka servers and visualize the data.

Prerequisites

- Java 1.6 or later is installed on your server.
- A project and a Metricstore are created. For more information, see Create a project and Create a Metricstore.

Limits

Only Linux Logtail V0.16.48 and later can be used to collect metric data from Kafka servers. If you installed an earlier version of Logtail on your server, you must update Logtail to a supported version.

Step 1: Create a Logtail configuration

- 1. Log on to the Log Service console.
- 2. In the Import Data section, click the Monitoring Data tab. Then, click Kafka Monitoring Data.
- You can also click View More and enter Kafka Monitoring Data in the search field of Import Data.
- 3. Select the project and Metricstore and click Next.
- 4. In the **Create Machine Group** step, create a machine group.
 - If a machine group is available, click Use Existing Machine Groups.
 - If no machine groups are available, perform the following steps to create a machine group. In this example, an Elastic Compute Service (ECS) instance is used.
 - a. Select the ECS instance on which you want to install Logtail. For more information, see Install Logtail on a Linux server. If Logtail is installed on the ECS instance, click **Complete Installation**.

() Note If you want to collect logs from self-managed clusters or servers that are provided by third-party cloud service providers, you must manually install Logtail. For more information, see Install Logtail on a Linux server.

- b. After Logtail is installed, click Complete Installation.
- c. Create a machine group.

For more information about how to create a machine group, see Create an IP address-based machine group or Create a custom identifier-based machine group.

5. Select the machine group from the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

① Important If you enable a machine group immediately after you create the machine group, the heartbeat status of the machine group may be FAIL. This issue occurs because the machine group is not connected to Log Service. To resolve this issue, you can click Automatic Retry. If the issue persists, troubleshoot the issue. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail client?.

6. In the Specify Data Source step, configure the following parameters. Then, click Next.

Parameter	Description
Configuration Name	The name of the Logtail configuration. You can enter a custom name.

Cluster Name	The name of the Kafka cluster. You can enter a custom name. After you configure this parameter, Log Service adds a <i>cluster=Cluster name</i> tag to the Kafka metric data that is collected by using the Logtail configuration.
	() Important Make sure that the cluster name is unique. Otherwise, data conflicts may occur.
Server List	 The information about the Kafka server. The information includes the following configuration items: Address: the address of the Kafka server. You can enter the IP address, hostname, or domain name of the server. Port: the port number of the Kafka server. Default value: 7777. You can add information about multiple Kafka servers based on your business requirements.
Custom Tags	The custom tags that are added to the collected Kafka metric data. The tags are key-value pairs. After you configure this parameter, Log Service adds the custom tags to the Kafka metric data that is collected by using the Logtail configuration.

Step 2: Configure JavaAgent

After the Logtail configuration is created, you must enable access to JMX data over HTTP. Log Service allows you to use Jolokia to access JMX data over HTTP. For more information, see Jolokia. You can download and load Jolokia based on the official documentation of Jolokia. You can also use Jolokia JavaAgent that is provided together with Logtail in Log Service. Jolokia JavaAgent is stored in /etc/ilogtail/telegraf/javaagent/jolokia-jvm.jar

You must configure the KAFKA_JVM_PERFORMANCE_OPTS environment variable on your Kafka server. For example, specify export KAFKA_JVM_PERFORMANCE_OPTS=-javaagent:/etc/ilogtail/telegraf/javaagent/jolokia-jvm.jar=port=7777 In this example, 7777 indicates the port number of your Kafka server. This port number must be the same as the port number that you specify in Step 1: Create a Logtail configuration.

? Note

By default, Jolokia JavaAgent listens only on the IP address 127.0.0.1 and allows requests only from the local host. If Logtail and your Java application are installed on different servers, you can add the **host=** field to the added script. This way, Jolokia JavaAgent can listen on other IP addresses. If you add **host=0.0.0.0**, Jolokia JavaAgent listens on all IP addresses. Example:

-javaagent:/tmp/jolokia-jvm.jar=port=7777,host=0.0.0.0

After you configure the settings, you must restart your Java application. If your Java application fails to restart, run the following command to connect Jolokia JavaAgent to a specified Java process. This way, the configuration immediately takes effect. Replace **PID** with the actual value.

This operation is used only for testing. You must complete the configuration based on the preceding descriptions. Otherwise, the Note configuration becomes invalid after your application restarts.

java -jar /etc/ilogtail/telegraf/javaagent/jolokia-jvm.jar --port 7777 start Process PID

If information similar to the following code is returned, the connection is successful:

Jolokia is already attached to PID 752

http://127.0.0.1:7777/jolokia/

After the connection is established, you can access the following URL to test connectivity:

curl http://127.0.0.1:7777/jolokia/

Sample response

{"request":{"type":"version"},"value":{"agent":"1.6.2","protocol":"7.2","config":

{"listenForHttpService":"true","maxCollectionSize":"0","authIgnoreCerts":"false","agentId":"30.**.**.186-752-5b091b5d-

(indemodified provide a constraint of a definition of a definition of a second a constraint of a second of a scoveryEnabled":"true","streaming":"true","canonicalNaming":"true","historyMaxEntries":"10","allowErrorDetails":"true","allowEnsReverseLookup":" ,"realm":"jolokia","includeStackTrace":"true","maxObjects":"0","useRestrictorService":"false","debugMaxEntries":"100"},"info": {"product":"tomcat","vendor":"Apache","version":"8.5.57"}},"timestamp":1602663330,"status":200}

FAO

How do I check whether Telegraf is collecting data as expected?

You can check the logs in the /etc/ilogtail/telegraf/telegraf.log file on your server. You can also collect the logs to Log Service and query the logs.

What to do next

· Query and analysis

After the configuration is complete. Telegraf collects metric data from Elasticsearch clusters, and Logial sends the metric data to a Metricstore in Log Service. You can query and analyze the metric data in the Metricstore. For more information, see Query and analyze time series data

Visualization

After the configuration is complete, Log Service automatically creates a dashboard named *kafka Monitoring_Cluster name* in the related project. You can visualize guery and analysis results on the dashboard. You can also configure alerts for the dashboard.

4.10.1.3.8. Collect metric data from Java applications or Tomcat servers

You can use Telegraf to collect metric data from Java applications or Tomcat servers. Then, you can use Logtail to send the metric data to a Metricstore in Log Service. This way, you can monitor the metric data of Java applications and Tomcat servers in a visualized manner. This topic describes how to collect metric data from Java applications by using Log Service and visualize the data.

Prerequisites

Logital V0.16.48 or later is installed on a Linux server. For more information, see Install Logital Logital on a Linux server.

· Java 1.6 or later is installed on the server.

Step 1: Create a Logtail configuration

- 1. Log on to the Log Service console.
- 2. In the Import Data section, click Java Application Monitoring Data.
- If you want to collect metric data from Tomcat servers, click Tomcat Monitoring Data.
- In the Specify Logstore step, select the project and the Metricstore that you want to use. Then, click Next. You can also click Create Now to create a project or a Metricstore. For more information, see Create a project and Create a Metricstore.
- In the Create Machine Group step, create a machine group.
 - If a machine group is available, click Use Existing Machine Groups.
 - If no machine groups are available, perform the following steps to create a machine group. In this example, an Elastic Compute Service (ECS) instance is used.
 - a. Select the ECS instance on which you want to install Logtail. For more information, see Install Logtail on a Linux server. If Logtail is installed on the ECS instance, click Complete Installation.

③ Note If you want to collect logs from self-managed clusters or servers that are provided by third-party cloud service providers, you must manually install Logtail. For more information, see Install Logtail on a Linux server.

- b. After Logtail is installed, click **Complete Installation**.
- c. Create a machine group.

For more information about how to create a machine group, see Create an IP address-based machine group or Create a custom identifier-based machine group.

5. Select the machine group from the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

① Important If you enable a machine group immediately after you create the machine group, the heartbeat status of the machine group may be FAIL. This issue occurs because the machine group is not connected to Log Service. To resolve this issue, you can click Automatic Retry. If the issue persists, troubleshoot the issue. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail client?.

6. In the Specify Data Source step, configure the parameters. The following table describes the parameters.

Parameter	Description
Configuration Name	Enter a name for the Logtail configuration.
Application Name	Enter the name of your Java application. After you configure this parameter, Log Service adds acluster= <application name=""> tag to the data that is collected. Note Make sure that the name of the application is unique. Otherwise, data conflicts may occur.</application>
Server List	Click the + icon to add a server on which your application resides. Address: the address of the server. Port: the port number of the server. You can specify a custom port number. The port number must be the same as the port number that you specify in Step 2: Configure JavaAgent. You can add multiple servers based on your business requirements.
Custom Tags	Create a custom tag in Custom Tags . The created tag is added to the data that is collected based on the Logtail configuration. You can use custom tags to identify the data that is collected based on different Logtail configurations in a Metricstore. You can click the + icon to create a custom tag. You can create multiple custom tags. The custom tags are added to all data that is collected based on the Logtail configuration.

Step 2: Configure JavaAgent

After the Logtail configuration is created, you must enable access to JMX data over HTTP. Log Service allows you to useJolokia to access JMX data over HTTP. You can download and load Jolokia based on the official documentation of Jolokia. You can also use Jolokia JavaAgent that is provided in Log Service together with Logtail. Jolokia JavaAgent is stored in /etc/ilogtail/telegraf/javaagent/jolokia-jvm.jar.

- If you want to collect metric data from Java applications, you must add the script -javaagent:/etc/ilogtail/telegraf/javaagent/jolokia-jvm.jar=port=7777 to the Java startup parameters.
- If you want to collect metric data from Tomcat servers, you must configure the JAVA_OPTS environment variable. For example, specify export JAVA_OPTS="-javaagent:/etc/ilogtail/telegraf/jolokia-jvm.jar=port=7777" . In this example, **7777** indicates the port number of the application server. This port number must be the same as the port number that you specify in Step 1: Create a Logtail configuration.

ONote By default, Jolokia JavaAgent listens only on the IP address 127.0.0.1 and allows requests only from the local host. If Logtail and your Java application are installed on different servers, you can add the host= field to the added script. This way, Jolokia JavaAgent can listen on other IP addresses. If you add host=0.0.0.0, Jolokia JavaAgent listens on all IP addresses. Example:

-javaagent:/tmp/jolokia-jvm.jar=port=7777,host=0.0.0.0

After you configure the settings, you must restart your Java application. If your Java application fails to restart, run the following command to connect Jolokia JavaAgent to a specified Java process. This way, the configuration immediately takes effect. Replace **PID** with the actual value.

() Note This command is used only for testing. You must complete the settings based on the preceding steps. Otherwise, the configuration becomes invalid after your application restarts.

java -jar /etc/ilogtail/telegraf/javaagent/jolokia-jvm.jar --port 7777 start PID

If the following output is returned, the connection is successful.

Jolokia is already attached to PID 752

http://127.0.0.1:7777/jolokia/

After the connection is established, you can access the following URL to verify the connection:

curl http://127.0.0.1:7777/jolokia/

Sample respons

{"request": {"type":"version"}, "value": {"agent":"1.6.2", "protocol":"7.2", "config":

{"listenForHttpService":"true", "maxCollectionSize":"0", "authIgnoreCerts":"false", "agentId":"30.43.124.186-752-5b091b5d-

instantion(tppervice.crue, maxcorrection):e.o., auchygorecetts.cruse, agentit.50.40.124.100-72-5009150d-jvm","debug":"false","agentType":"jvm","policyLocation":"classpath:\/jolokia-access.wml","agentContext":"\/jolokia","serializeException":"false","mimeType":"text\/plain","maxDepth":"15","authMode":"basic","authMatch":"any scoveryEnabled":"true","streaming":"true","canonicalNaming":"true","historyMaxEntries":"10","allowErrorDetails":"true","allowDnsReverseLookup": ,"realm":"jolokia","includeStackTrace":"true","maxObjects":"0","useRestrictorService":"false","debugMaxEntries":"100"},"info": {"product":"tomcat","vendor":"Apache","version":"8.5.57"}},"timestamp":1602663330,"status":200}@

FAQ

How do I check whether Telegraf is collecting data as expected?

You can check the logs of the /etc/ilogtail/telegraf/telegraf.log file on your server. You can use Log Service to collect this log file and search for the required information in Log Service.

What to do next

Ouerv and analyze data

After you configure the settings, Telegraf uses Logtail to upload collected metric data to the specified Metricstore in Log Service. You can query and analyze the data on the Query & Analysis page of the Metricstore. For more information, see Query and analyze time series data Visualize guery and analysis results

You can create a time series chart and add the chart to a dashboard to view the query and analysis results. You can also configure alert rules.

4.10.1.3.9. Collect metric data from NGINX servers

NGINX provides a built-in status page that allows you to monitor the status of NGINX metric data. You can use Telegraf to collect metric data from NGINX servers, and then use Logiail to send the metric data to a Metricstore in Log Service. This way, you can monitor metric data of NGINX servers in a visualized manner. This topic describes how to collect metric data from a NGINX server by using Log Service and visualize the data.

Prerequisites

Logtail V0.16.50 or later is installed on a Linux server. For more information, see Install Logtail on a Linux server.

Step 1: Configure the NGINX status module

1. Run the following command to check whether NGINX has the status module. For more information, see Module ngx_http_status_module.

nginx -V 2>&1 | grep -o with-http_stub_status_module with-http_stub_status_module

If the message with-http_stub_status_module is returned, NGINX has the status module.

```
2. Configure the NGINX status module.
```

Configure the status module in the NGINX configuration file. By default, this file is stored in the /etc/nginx/nginx.conf directory. Use the following sample code to configure the status module. For more information, see NGINX status

```
location /private/nginx status
 stub_status on;
 access_log off;
 allow 192.0.2.1;
 deny all;
```

• /private/nginx status indicates the URI of the NGINX status module. Replace the value with the actual URI.

• allow 192.0.2.1 indicates that only the IP address 192.0.2.1 is allowed to access the NGINX status module. Replace the value with the actual IP address

3. Run the following command to check whether the server on which Logtail is installed can access the NGINX status module:

\$curl http://192.0.2.1/private/nginx status

If the following message is returned, the NGINX status module is configured:

```
Active connections: 1
server accepts handled requests
2507455 2507455 2512972
Reading: 0 Writing: 1 Waiting: 0
```

Step 2: Import data

- 1. Log on to the Log Service console.
- 2. In the Import Data section, click NGINX Monitoring.
- 3. In the Specify Logstore step, select the project and the Metricstore that you want to use. Then, click Next.
- You can also click Create Now to create a project or a Metricstore. For more information, see Create a project and Create a Metricstore.
- 4. In the Create Machine Group step, create a machine group
 - If a machine group is available, click Use Existing Machine Groups.
 - If no machine groups are available, perform the following steps to create a machine group. In this example, an Elastic Compute Service (ECS) instance is used.
 - a. Select the ECS instance on which you want to install Logtail. For more information, see Install Logtail on a Linux server. If Logtail is installed on the ECS instance, click Complete Installation.

(?) Note If you want to collect logs from self-managed clusters or servers that are provided by third-party cloud service providers, you must manually install Logtail. For more information, see Install Logtail on a Linux server

b. After Logtail is installed, click Complete Installation.

c. Create a machine group

For more information about how to create a machine group, see Create an IP address-based machine group or Create a custom identifier-based

5. Select the machine group from the Source Server Groups section and move the machine group to the Applied Server Groups section. Then, click Next.

Important If you enable a machine group immediately after you create the machine group, the heartbeat status of the machine group may be FAIL. This issue occurs because the machine group is not connected to Log Service. To resolve this issue, you can click Automatic Retry. If the issue persists, troubleshoot the issue. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail client?

6. In the Specify Data Source step, configure the parameters and click Next. The following table describes the parameters.

Parameter	Description
Configuration Name	Enter a name for the Logtail configuration.
Cluster Name	Enter the name of the cluster. After you configure this parameter, Log Service adds a <i>cluster=<cluster name=""></cluster></i> tag to the data that is collected.
	(?) Note Make sure that the cluster name is unique. Otherwise, data conflicts may occur.
	Click the + icon to add the NGINX server and configure the following parameters:
	Address: the endpoint of the NGINX server.
Commentiat	• Port : the port number of the NGINX server.
Server List	 Path: the URI of the NGINX status module. Example:/private/nginx_status. For information about how to configure the NGINX status module, see Step 1: Configure the NGINX status module
	You can add multiple NGINX servers based on your business requirements.
Custom Tags	Create a custom tag in Custom Tags . The created tag is added to the data that is collected based on the Logtail configuration. You can use custom tags to identify the data that is collected based on different Logtail configurations in a Metricstore.
	You can click the + icon to create a custom tag. You can create multiple custom tags. The custom tags are added to all data that is collected based on the Logtail configuration.

FAO

How do I check whether Telegraf is collecting data as expected?

You can check the logs of the /etc/ilogtail/telegraf/telegraf.log file on your server. You can use Log Service to collect this log file and search for the required information in Log Service.

What to do next

• Query and analyze data

After you configure the settings, Telegraf uses Logtail to upload collected metric data to the specified Metricstore in Log Service. You can query and analyze the data on the Query & Analysis page of the Metricstore. For more information, see Query and analyze time se Visualize query and analysis results

You can create a time series chart and add the chart to a dashboard to view the query and analysis results. You can also configure alert rules.

4.10.1.3.10. Collect metric data from NVIDIA GPUs

you can monitor NVIDIA GPUs in a visualized manner. This topic describes how to use Log Service to collect metric data from NVIDIA GPUs and visualize the data.

Prerequisites

A project and a Metricstore are created. For more information, see Create a project and Create a Metricstore.

Limits

Only Linux Logtail V0.16.50 and later can be used to collect metric data from NVIDIA GPUs. If you have installed an earlier version of Logtail on your server, you must update Logtail to a supported version.

Step 1: Install an NVIDIA GPU driver

Log Service uses the nvidia-smi command to collect GPU information. The command is included in a GPU driver. You must install a GPU driver before you can use Log Service to collect metric data from NVIDIA GPUs. If you use a GPU-accelerated instance of Elastic Compute Service (ECS), the driver is installed by default. In this case, you can skip this step.

Step 2: Create a Logtail configuration

- 1. Log on to the Log Service console.
- In the Import Data section, click the Monitoring Data tab. Then, click Nvidia GPU Monitoring. You can also click View More and enter Nvidia GPU Monitoring in the search field of Import Data
- 3. Select the project and Metricstore and click Next.
- 4. In the Create Machine Group step, create a machine group.
 - If a machine group is available, click Use Existing Machine Groups.
 - If no machine groups are available, perform the following steps to create a machine group. In this example, an Elastic Compute Service (ECS) instance is used
 - a. Select the ECS instance on which you want to install Logtail. For more information, see Install Logtail on a Linux server. If Logtail is installed on the ECS instance, click Complete Installation.

Note If you want to collect logs from self-managed clusters or servers that are provided by third-party cloud service providers, you must manually install Logtail. For more information, see Install Logtail on a Linux server

b. After Logtail is installed, click **Complete Installation**.

c. Create a machine group.

For more information about how to create a machine group, see Create an IP address-based machine group or Create a custom identifier-based machine group.

5. Select the machine group from the **Source Server Groups** section and move the machine group to the **Applied Server Groups** section. Then, click **Next**.

Important If you enable a machine group immediately after you create the machine group, the heartbeat status of the machine group may be FAIL. This issue occurs because the machine group is not connected to Log Service. To resolve this issue, you can click Automatic Retry. If the issue persists, troubleshoot the issue. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail client?.

6. In the Specify Data Source step, configure the parameters. The following table describes the parameters. Click Next.

Parameter	Description
Configuration Name	The name of the Logtail configuration. You can enter a custom name.
Cluster Name	The name of the NVIDIA GPU cluster. You can enter a custom name. After you configure this parameter, Log Service adds a <i>cluster=Cluster name</i> tag to the NVIDIA GPU metric data that is collected by using the Logtail configuration. ① Important Make sure that the cluster name is unique. Otherwise, data conflicts may occur.
Nvidia SMI Path	The directory in which nvidia-smi is installed. Default value:/usr/bin/nvidia-smi.
Custom Tags	The custom tags that are added to the collected NVIDIA GPU metric data. The tags are key-value pairs. After you configure this parameter, Log Service adds the custom tags to the NVIDIA GPU metric data that is collected by using the Logtail configuration.

FAQ

How do I check whether Telegraf is collecting data as expected?

You can check the logs in the /etc/ilogtail/telegraf/telegraf.log file on your server. You can also collect the logs to Log Service and query the logs.

What to do next

Query and analysis

After the configuration is complete, Telegraf collects metric data from Elasticsearch clusters, and Logtail sends the metric data to a Metricstore in Log Service. You can query and analyze the metric data in the Metricstore. For more information, see Query and analyze time series data.

Visualization

After the configuration for NVIDIA GPUs is complete, Log Service automatically creates a dashboard named NVIDIA_GPU Monitoring_Cluster name in the project that is used. In the dashboard, you can perform various operations. For example, you can view query results and configure alerts.

4.10.1.4. Collect metric data from Prometheus

4.10.1.4.1. Collect metric data from Prometheus by using the Remote Write

Protocol

Prometheus is a cloud native application that you can use to collect and monitor metric data of various software and systems. This topic describes how to import metric data from Prometheus to Log Service. This topic also describes how to use Log Service to analyze and monitor the data.

Prerequisites

- A Metricstore is created. For more information, see Create a Metricstore.
- Prometheus is installed. For more information, see GETTING STARTED.
- Data collection rules are configured in Prometheus. For more information, see scrape_config.

Procedure

Log Service supports the Remote Write Protocol. You can use the remote write feature of Prometheus to import metric data to Log Service. Before you can use the remote write feature, you must perform the following steps to enable the feature in Prometheus:

- 1. Log on to the server on which Prometheus is installed.
- 2. Open the configuration file and configure the parameters based on your business scenario. The following table describes the parameters. For more information, see remote_write.

```
url: https://{project}.{sls-endpoint}/prometheus/{project}/{metricstore}/api/vl/write
basic_auth:
    username: access-key-id
    password: access-key-secret
queue_config:
    batch_send_deadline: 20s
    capacity: 20480
    max_samples_per_send: 2048
    min_backoff: 5s
    max_samples_per_send: 2048
    min_backoff: 100ms
    min_shards: 100
Parameter Description
```

url	The URL of the Metricstore in Log Service. The URL must be in the following format: https://project}.{sls-end point}/prometheus/{project}}{metricstore}}api/v1/write. Take note of the following variables: • {sls-endpoint}: the Log Service endpoint. For more information, see Obtain an endpoint in Log Service Developer Guide. • {project}: the project that you created. • {metricstore}: the Metricstore that you created. • {metricstore}: the Metricstore that you created. • Important • If you use an Alibaba Cloud internal network, we recommend that you use an internal Log Service endpoint. • To ensure secure transmission, you must use HTTPS.
basic_auth	 The authentication information. If data is written to Log Service over the Remote Write Protocol, basic authentication is required. Take note of the following parameters: username: the AccessKey ID of your Apsara Stack tenant account. password: the AccessKey secret of your Apsara Stack tenant account. We recommend that you use the AccessKey pair of a RAM user that is granted only the write permissions on the Log Service project.
queue_config	<pre>queue_config specifies the cache and retry policies for data writes. To reduce invalid network requests, setmin_backoff to a value that is greater than or equal to100ms and set max_backoff to a value that is greater than or equal to5s. If you want to collect a large amount of data from Prometheus, use the following settings forqueue_config: batch_send_deadline: 20s capacity: 20480 max_backoff: 5s max_samples_per_send: 2048 min_backoff: 100ms min_shards: 100</pre>

 Check whether data is imported to Log Service.
 After you configure Prometheus, you can use the preview feature in the Log Service console to check whether data is imported to Log Service. i. Log on to the Log Service console.

ii. In the Projects section, click the project that you want to manage.

iii. Choose Time Series Storage > Metricstore. On the Metricstore tab, find the Metricstore that you want to manage and choose 🔀 >

Consumption Preview. If data is displayed in the Consumption Preview panel, Prometheus is correctly configured.

onsumption Pre	eview			
te		Shard:0	✓ 15 Minutes ∨	Preview
Log preview is only u through keywords, e	used to check whether log da nable log index.	ata is uploaded succ	essfully. If you want to sear	ch logs
Time/Source	Content			
2011-110-140 542-112-101 1712-12-108, 1605	_3040_10000 1.001_3040	urf56chao kodge ana 1021447900		1.10
2011-110-00 140-00-00 170-00-00-005	_labels_hostnar m_load5 _5me_m	ne#\$#ichao-lest[pi ana 1601447908		syste
2001-09-00 54:00:00 170-0008-005	iabelshostnar m_load15time_	ne#\$#lichao-lest[pr nane 160144790	1000 0.00 00 _0000 0000000 _0000_0	syste
2007/09.00 54/00/00 170-0000.500	labelshostnar m_boot_timetim 0167e+09	ne#S#lichao-lest[pf se_nane160144		1.58796
2000-00-00 54100-00 571-00-00 590	labelshostnar ourltime_nano	ne#\$#ichao.test(ipt 1601447906165		_tpu_t
				Clo

What to do next

After metric data is collected from Prometheus, you can perform the following operations on the data:

- Query and analyze Prometheus metric data in Log Service. For more information, see Query and analyze time series data.
- Visualize Prometheus metric data in Grafana. For more information, see Send time series data from Log Service to Grafana.

4.10.1.4.2. Collect metric data from Prometheus by using a Logtail plug-in

Log Service allows you to collect various types of Prometheus metrics by using a Logtail plug-in. The metrics include Prometheus-formatted metrics from Node Exporter and Kafka Exporter, and Prometheus metrics that are collected from applications. This topic describes how to create a Logtail configuration in the Log Service console to collect metric data from Prometheus.

Prerequisites

A Metricstore is created. For more information, see Manage a Metricstore.

Procedure

() Important A Logital plug-in supports only one Logital configuration for Prometheus. If more than one configuration exists, Logital uses a random configuration.

- 1. Log on to the Log Service console.
- 2. In the Import Data section, click Prometheus Metric Scrape.
- 3. In the Specify Logstore step, select the project and the Metricstore that you want to use. Then, click Next.
- 4 Create a machine group
 - If a machine group is available, click Use Existing Machine Groups.
 - If no machine groups are available, perform the following steps to create a machine group. In this example, an Elastic Compute Service (ECS) instance is used.
 - a. On the ECS Instances tab, select Manually Select Instances. Then, select the ECS instance that you want to use and click Execute Now. For more information, see Obtain an endpoint in Log Service Developer Guide.

③ Note If you want to use a server in a self-managed cluster or a server that is deployed on a third-party cloud, you must manually install Logtail V0.16.66 or later on the Linux server. For more information, see Install Logtail on a Linux server.

- b. After Logtail is installed, click Complete Installation.
- c. In the Create Machine Group step, configure the Name parameter and click Next.
- Log Service allows you to create IP address-based machine groups and custom ID-based machine groups. For more information, see Create an IP ine group and Create a custom identifier-based machine group.
- 5. Select the machine group from the Source Server Groups section and move the machine group to the Applied Server Groups section. Then, click Next.

Important If you enable a machine group immediately after you create the machine group, the heartbeat status of the machine group may be FAIL. This issue occurs because the machine group is not connected to Log Service. To resolve this issue, you can click Automatic Retry. If the issue persists, see What do I do if a Logtail machine group has no heartbeats?

6. In the Specify Data Source step, configure the Config Name and Plug-in Config parameters. Then, click Next. Plug-in Config includes the inputs and processors parameters. Log Service provides a template for the inputs parameter. The template includes only the global and scrape_configs sections.

• inputs: specifies the collection configurations. This parameter is required. Configure the inputs parameter based on your data source.

- (!) Important
 - You can configure fields only in the global and scrape_configs sections regardless of whether you collect Prometheus-formatted metrics or Prometheus metrics. For more information, see Prometheus configuration.
 - You can specify only one type of data source in the inputs parameter.

processors: specifies the processing method. This parameter is optional. If you want to append custom fields, such as the IP address of the server on which Logtail is installed and the hostname of the server, to the collected metric data, you must turn on **Use Advanced Edit Mode** to add **processors** settings. In this case, the processor_appender plug-in must be used. Example:

{			
	"p	r	ocessors":[
		{	
			"type":"processor_appender",
			"detail": {
			"Key": "labels",
			"Value": " host#\$#{{host}} ip#\$#{{ip}}"
			"SortLabels": true
			}
		}	
]		
}			

For more information, see Add log fields.

What to do next

Ouery and analyze data

After metric data is collected, you can query and analyze the data on the Query & Analysis page of the Metricstore. For more information, see Query and analyze time series dat

- · Visualize query and analysis results in Log Service
- You can create a time series chart and add the chart to a dashboard to view the query and analysis results. You can also configure alert rules.
- Visualize data on Grafana

Log Service allows you to send time series data to Grafana for visualization. For more information, see Send time series data from Log Service to

4.10.2. Query and analysis

4.10.2.1. Overview of query and analysis of time series data

This topic describes the syntax and limits of query and analysis on time series data.

- Log Service supports the following types of syntax for the query and analysis of time series data:
- SQL: You can use the SQL syntax to query and analyze time series data based on the encoding format of the data.
- Combination of SQL and PromQL: This combination allows you to query and analyze time series data in an efficient manner. The combination is
 implemented by using nested queries. PromQL is the query language that is provided by Prometheus. For more information, see Prometheus
 documentation.

SQL

Examples of SQL-based query statements:

• Query and analyze all data from a Metricstore.

*| SELECT * FROM "my_metric_store.prom" WHERE __name__ != ''

• Query the data in which the value of the _labels_, 'domain' field is www.example.com and obtain the sum of the values of the _value_ field.

*| SELECT sum(__value__) FROM "my_metric_store.prom" WHERE element_at(__labels__, 'domain')='www.example.com'

• Query the data in which the value of the _labels_, 'domain' field is www.example.com, obtain the sum of the _value_ field, and aggregate the data by hour.

```
*! SELECT sum(_value_),date_trunc('hour', __time_nano_/1000000) as t
FROM "my_metric_store.prom"
WHERE element_at(_labels_, 'domain')='www.example.com'
GROUP BY t
ORDER BY t DESC
```

Description:

• The SQL syntax for time series data is the same as the SQL syntax for log data. For more information, see Log analysis overview. When you query and analyze time series data by using the SQL syntax, the table name in a FROM clause must be **{metrics_store_name}.prom**. *{metrics_store_na me}* specifies the name of the Metricstore that you created.

```
NoteYou must enclose the table name in double quotation marks ("").
```

• You can use the element_at() function to obtain the value of a key from the _labels_ field. Example: element_at(_labels_, 'key').

• For more information about the table structure, see Metricstore.

Combination of SQL and PromQL

If you use the combination of SQL and PromQL, you can use security check functions and advanced features, such as the machine learning feature.

() Important

- If you use the combination of SQL and PromQL, the table name in a FROM clause must be metrics.
- · For information about the API endpoints and descriptions of PromQL functions, see Prometheus documentation.

The following table describes the PromQL functions that are supported by Log Service. Among the functions, the promql_query, promql_labels, promql_label_values, and promql_series functions can be invoked only on the Query & Analysis page of a Metricstore.

Function	Description	Example
promql_query(string)	Evaluates an instant query on the data at the point in time that is the closest to the end time of the query time range. This function is equivalent to the /query API of Prometheus. Parameter setting: query= <string> .</string>	* SELECT promql_query('up') FROM metrics
promql_query_range(string, string)	Evaluates a query on the data over a specified period of time. This function is equivalent to the /query_range API of Prometheus. Parameter settings: query= <string> and step=<duration>.</duration></string>	* SELECT promql_query_range('up', '5m') FROM metrics
promql_labels()	Returns all label keys.	* SELECT promql_labels() FROM metrics
promql_label_values(string)	Returns the values of a label.	* SELECT promql_label_values('name_') FROM metrics
promql_series(string)	Returns the time series that is matched.	* SELECT promql_series('up') FROM metrics

A PromQL function is similar to a user-defined table generating function (UDTF) and returns a table.

The following table describes the schema of a table that is returned by the	promql_query(string)	or	<pre>promql_query_range(string, string)</pre>
function.			

Field	Data type	Description

metric	varchar	The metric name of the time series. If a GROUP BY clause is included in the query statement, this field may be empty.
labels	map <varchar, varchar=""></varchar,>	The labels. The value is a map.
time	bigint	The time.
value	double	The value that represents a point in time.

• The following table describes the schema of a table that is returned by the promql_labels() or promql_label_values(string) function.

Field	Data type	Description
label	varchar	Label Key

• The following table describes the schema of a table that is returned by the promql_series(string) function.

Field	Туре	Description
series	map <varchar, varchar=""></varchar,>	The time series.

Limits

- A Metricstore supports only the query API endpoints of Prometheus, such as /query and /query_range. Other API endpoints, such as /admin, /alerts, and /rules, are not supported.
- If you use the combination of SQL and PromQL for query and analysis, a maximum of 11,000 points in time can be returned for a query.
- If you use the combination of SQL and PromQL for query and analysis, the metric name and label that you specify must comply with the naming conventions. For more information, see Metricstore.

4.10.2.2. Query and analyze time series data

This topic describes how to query and analyze time series data in a Metricstore and how to specify a legend format for a time series chart.

Prerequisites

Time series data is collected. For more information, see Collect time series data.

Procedure

? Note

Only the combination of the SQL syntax and the PromQL syntax is supported on the query page of a Metricstore. If you want to use the standard SQL syntax, click the **Search & Analyze** button in the upper-right corner of the query page to go to the Search & Analysis page of the Metricstore.

1. Log on to the Log Service console

- 2. In the Projects section, click the project that you want to manage.
- 3. Choose Time Series Storage > Metricstore. On the Metricstore tab, click the Metricstore that you want to manage.
- 4. In the upper-right corner of the page that appears, click 15 Minutes (Relative) and specify a time range for data query and analysis.

You can select a relative time or a time frame. You can also specify a custom time range.

? Note

The query and analysis results may contain time series data that is generated 1 minute earlier or 1 minute later than the specified time range.

5. On the Query Statements tab, query and analyze the time series data.

You can use the following methods to query and analyze time series data:

 $\circ~$ Enter a query statement in the field next to the Metricstore name and click $\ensuremath{\textbf{Preview}}.$

You can click the Add Query Statement button or the 🗐 icon to add or copy a query statement. Then, enter the query statement and click Preview

- The results of multiple query statements are displayed in the same time series chart.
- Select a metric from the Metrics drop-down list. A query statement is automatically generated. Then, click Preview.
- You can modify the query statement that is generated.

Specify a legend format for a time series chart

After you execute a query statement on the Query Statements tab, you can specify a legend format for the time series chart.

The default legend for each time series consists of a metric name and labels. You can change the legend to a label value by using a magic variable. The format is **{Label key}}**. For example, the label of a time series chart is **{ip="192.0.2.1"}**. If you enter **{{ip}}** in the **Legend Format** field, the legend of the time series chart changes to **192.0.2.1**.

Configure a placeholder variable

Log Service allows you to specify placeholder variables in query statements. For example, you can configure a drill-down event for Chart A to redirect to the dashboard on which Chart B is located. After you configure a drill-down event for Chart A, the variable that you click to trigger the drill-down event and execute the query statement of Chart B is replaced by the placeholder variable specified in Chart B. To trigger the drill-down event, you must click the variable that you configured for Chart A. The format of a placeholder variable is $\{ \text{Variable name} \mid \text{Default value} \} \}$. For example, you can set $\text{hoste-"$}(\text{host}^-, *)^*$ to $\text{hoste-"}(\text{host}^-, *)^*$.

Example: In the following query statement, set the values of the **host**, **url**, **method**, **status**, and **proxy_upstream_name** fields to placeholder variables. The following figure shows the result.

* | select promql_query_range('sum(sum_over_time(pv:host:status:method:upstream_name:upstream_status:url{host=~"^.*", url=~".*\$, method=~".*", status=~".*", proxy_upstream_name=~".*"}[lm]))') from metrics limit 10000

nginx-ingress-m V	1 * I select	
	<pre>promql_query_range('sum(sum_over_time(pv:host:status:method:upstream_name:upstream_status:url{host=-'\${{host ^.*}} ", url=-'\${{url*}}, method=-'\${{method!.*}}, status=- \${{status!.*}}, proxy_upstream_name'\${{service!.*}}"}[lm] offset 1d))') from metrics limit 10000</pre>	ł

Related operations

Operation	Description
Copy a query statement	On the Query Statements tab, click the 🗐 icon to copy the specified query statement.
View raw data	On the Query Statements tab, click the \odot icon to view raw time series data.
Configure the properties of a time series chart	On the Properties tab, configure the properties of the time series chart. For more information, seeConfigure a time series chart.
Configure Interactive Behavior	Drill-down events are important for data analysis. You can use drill-down events to switch between the levels of data dimensions and the analysis granularities to obtain more detailed information. For more information, see Drill-down events.
Add query and analysis results to a dashboard	Click Add to New Dashboard to add the query and analysis results to a dashboard.
Create an alert rule based on query statements	Click Save as Alert to create an alert rule for the query and analysis results. For more information, see <u>Configure an alert rule</u> .
Refresh data	 You can manually refresh the time series data in a Metricstore or enable the automatic refresh feature. In the upper-right corner of the query page, chooseRefresh > Once to refresh the time series data in the Metricstore. In the upper-right corner of the query page, chooseRefresh > Auto Refresh and select a refresh interval. The time series data in the Metricstore is automatically refreshed based on the selected interval. The interval can be 15 seconds, 60 seconds, 5 minutes, or 15 minutes.
Share query and analysis results	In the upper-right corner of the query page, click Share to copy the URL of the page. You can send the URL to other users who have the permissions to query the time series data in the Metricstore. The query and analysis results that are displayed when you copy the URL are also displayed when a user visits the URL.
Go to the Search & Analysis page	In the upper-right corner of the query page, click Search & Analyze .

4.10.3. Visualization

4.10.3.1. Configure a time series chart

This topic describes how to configure a time series chart.

Background information

Time series charts are designed for Metricstores. A time series chart displays the results of one or more queries on data that is collected from Prometheus and stored to a Metricstore.

Procedure

- 1. Log on to the Log Service console
- 2. In the Projects section, click the project that you want to manage.
- 3. Choose Time Series Storage > Metricstore. On the Metricstore tab, click the Metricstore that you want to manage.
- 4. Click the **Properties** tab and configure the parameters of the time series chart. The following section describes the parameters.

Parameter	Description
Fill Missing Data	If you turn on Fill Missing Data , Log Service automatically generates substitutes for missing samples in a time series.

Y-axis Minimum Value	The minimum value allowed for the y-axis.		
Y-axis Maximum Value	The maximum value allowed for the y-axis.		
Format Left Y-axis	The format of the left y-axis.		
X-axis Scale Density	The scale density of the x-axis. Valid values: 3 to 30.		
Line Type	The type of the lines in the time series chart. Valid values: Straight Line and Curve.		
Show Points	If you turn on Show Points , sample values are displayed in the time series chart.		
Margin	The distance between an axis and a border of the time series chart.		
Display Inspection Results	If you turn on Display Inspection Results , you can configure event behavior action for associate metrics.		

5. On the Query Statements tab, query and analyze the time series data.

You can click Preview Raw Data in the upper-right corner of the page to view the collected time series data. Example:

labels_:hostname#\$#hostname1|ip#\$#192.0.2.0_time_nano_:164430967100000000_value_:52.71_name_:cpu_uti1 . Log Service generates a time series chart based on the collected time series data and the metrics that you select. For example, if you want to query the CPU utilization of different hosts, you can select the **cpu_util** metric. Then, Log Service displays a time series chart for the CPU utilization of different hosts.

4.10.3.2. Send time series data from Log Service to Grafana

Metricstores of Log Service are compatible with the query API of Prometheus. You can send data from Log Service to Grafana and visualize the data in Grafana. This topic describes how to connect Log Service as a Prometheus data source to Grafana.

Prerequisites

- Grafana is installed. For more information, see Install Grafana.
- Time series data is imported to Log Service. For more information, see Collect metric data from Prometheus.

Connect Log Service to Grafana

- 1. Log on to Grafana.
- 2. In the left-side navigation pane, choose **Configuration > Data Sources**.
- 3. On the Data Sources tab, click Add data source.
- 4. Move the pointer over the Prometheus card and click Select.
- 5. On the **Settings** tab, configure the parameters. The following table describes the parameters.

Parameter	Description		
Name	Specify a name for the data source based on your business requirements. Example: Prometheus-01.		
НТТР	 URL: Enter the URL of the Metricstore in the https://{project}.{sls-enpoint}/prometheus/{project}/{metricstore} format. Replace {sls-enpoint} with the Log Service endpoint in the region where the project resides. To obtain the list of Log Service endpoints in different regions, see Obtain an endpoint in Log Service Developer Guide. Replace {project} and {metricstore} with the actual project name and Metricstore name. Example:https://sls-prometheus-test.cn-hangzhou.log.allyuncs.com/prometheus/sls-prometheus-test/prometheus. Note To ensure the security of transmissions, use https • Whitelisted Cookies: Add a whitelist. This parameter is optional. 		
Auth	Turn on Basic auth .		
Basic Auth Details	 User: Enter the AccessKey ID of your Apsara Stack tenant account. Password: Enter the AccessKey secret of your Apsara Stack tenant account. We recommend that you use the AccessKey pair of a RAM user that is granted only the read-only permissions on the specified project. 		

6. Click Save & Test.

Import a Log Service dashboard template to Grafana

Perform the following steps to import a Log Service dashboard template to Grafana:

1. Copy the ID of a template.

- i. Go to the Dashboards page of Grafana.
- ii. Click the template that you want to import.
- iii. On the right side of the page, click **Copy ID to Clipboard**.
- 2. Log on to Grafana.
- 3. In the left-side navigation pane, choose **Create > Import**.
- 4. In the Grafana.com Dashboard field, paste the template ID that you copied in Step 1.
- Then, click a blank area to go to the page for data source configuration.
- 5. Configure the data source.

In this step, configure the parameters based on the data source that you added. For more information, see Connect Log Service to Grafana. The parameters vary based on the dashboard template. You can also configure the **telegraf** parameter and **host** parameter based on your business requirements.

6. Click Import.

Access Log Service by using the query API of Prometheus

Log Service is compatible with the query API of Prometheus. You can configure Log Service as a Prometheus data source in Grafana or use the Prometheus API to access Log Service. The following table describes the API operations that are supported.

Operation	Example	
Instant queries	GET /api/vl/query POST /api/vl/query	
Range queries	GET /api/vl/query_range POST /api/vl/query_range	
Getting label names	GET /api/v1/labels POST /api/v1/labels	
Querying label values	GET /api/v1/label/ <label_name>/values</label_name>	
Finding series by label matchers	GET /api/vl/series POST /api/vl/series	

4.11. RAM

4.11.1. Permissions required by a RAM user to manage Log Service

resources

This topic describes the permissions that you must grant a Resource Access Management (RAM) user to manage Log Service resources.

If you want to use a RAM user to perform operations on Log Service resources in the console, you must contact the administrator to grant the RAM user the following permissions:

Data management permissions on a project

You can grant users the permissions to access the instances of specified cloud services, and view and modify the data-related permissions of specified users. The following list describes how to grant permissions to a RAM user:

- i. Select Log Service for Product Type.
- ii. Select a project and grant the required permissions to the RAM user on the project.
- iii. Find the RAM user and click Authorize in the Actions column. Then, select the permissions that you want to grant the RAM user and complete the authorization.

For more information, see Apsara Uni-manger > User Guide > Enterprise > Permissions > Data Permissions >

• Fine-grained permissions based on a custom RAM role and policies

For more information about how to create a RAM user, create a policy, and perform authorization, see **Apsara Uni-manger > User Guide >** Enterprise > Permissions > Role Permissions.

Log Service allows you to configure custom policies that grant permissions to RAM users. You can attach the policies to the RAM users by using the Log Service console and API operations. For more information, see Use custom policies to grant permissions to a RAM user.

4.11.2. Use custom policies to grant permissions to a RAM user

This topic describes how to use custom policies to grant permissions to a RAM user. In the Resource Access Management (RAM) console, you can grant permissions to the RAM users that belong to your Apsara Stack tenant account.

Background information

In terms of data security, we recommend that you follow the principle of least privilege (PoLP) when you grant permissions to RAM users. You must grant the read-only permissions on the project list to RAM users. Otherwise, the RAM users cannot view the projects in the project list.

Use the RAM console to grant permissions to a RAM user

- The read-only permissions on projects
 - For example, you want to use your Apsara Stack tenant account to grant the following permissions to a RAM user:
 - $\circ\;$ The permissions to view the project list of the Apsara Stack tenant account

 The read-only permissions on the projects that are specified by the Apsara Stack tenant account Use the following policy:

Cloud Defined Storage

```
{
       "Version": "1",
       "Statement": [
         {
           "Action": ["log:ListProject"],
           "Resource": ["acs:log:*:*:project/*"],
"Effect": "Alow"
          },
         {
           "Action": [
             "log:Get*",
             "log:List*"
           ],
           "Resource": "acs:log:*:*:project/Project name/*",
           "Effect": "Allow"
         }
      ]
    }
• The read-only permissions on a specified Logstore and the permissions to save a search and use the saved search
  For example, you want to use your Apsara Stack tenant account to grant the following permissions to a RAM user:
 • The permissions to view the project list of the Apsara Stack tenant account
  • The read-only permissions on a specified Logstore and the permissions to save a search and use the saved search
  Use the following policy.
    ? Note
   If the content of the Resource element in a policy does not end with an asterisk (*), the RAM user can access only the specified resource of the current resource type. If the content of the Resource element ends with an asterisk (*), the RAM user can access all resources of the current
    resource type. Other resources are represented by an asterisk (*).
   {
      "Version": "1",
      "Statement": [
        {
          "Action": [
             "log:ListProject"
          ],
          "Resource": "acs:log:*:*:project/*",
          "Effect": "Allow"
        }.
       {
          "Action": [
            "log:List*"
          "Resource": "acs:log:*:*:project/Project name/logstore/*",
          "Effect": "Allow"
        },
          "Action": [
           "log:Get*",
           "log:List*"
          ],
          "Resource": [
            "acs:log:*:*:project/Project name/logstore/Logstore name>"
          1,
          "Effect": "Allow"
        },
        {
          "Action": [
            "log:List*"
          1,
           "Resource": [
           "acs:log:*:*:project/Project name/dashboard",
            "acs:log:*:*:project/Project name/dashboard/*"
          "Effect": "Allow"
        },
        {
          "Action": [
            "log:Get*",
            "log:List*",
           "log:Create*"
          ],
          "Resource": [
            "acs:log:*:*:project/Project name/savedsearch",
            "acs:log:*:*:project/Project name/savedsearch/*"
          ],
          "Effect": "Allow"
     ]
   }
```

• The read-only permissions on a specified Logstore and the permissions to view all saved searches and dashboards in a project

For example, you want to use your Apsara Stack tenant account to grant the following permissions to a RAM user:

 $\circ\;$ The permissions to view the project list of the Apsara Stack tenant account

• The read-only permissions on a specified Logstore and the permissions to view all saved searches and dashboards in a project Use the following policy:

```
"Version": "1",
  "Statement": [
    {
      "Action": [
        "log:ListProject"
      ],
      "Resource": "acs:log:*:*:project/*",
     "Effect": "Allow"
    },
    {
      "Action": [
        "log:List*"
     1,
      "Resource": "acs:log:*:*:project/Project name/logstore/*",
      "Effect": "Allow"
    },
    {
      "Action": [
       "log:Get*",
        "log:List*"
      ],
      "Resource": [
       "acs:log:*:*:project/Project name/logstore/Logstore name"
      "Effect": "Allow"
    },
    {
      "Action": [
       "log:Get*",
       "log:List*"
      ],
      "Resource": [
       "acs:log:*:*:project/Project name/dashboard",
        "acs:log:*:*:project/Project name/dashboard/*"
      1,
      "Effect": "Allow"
    },
      "Action": [
        "log:Get*",
       "log:List*"
      ],
      "Resource": [
        "acs:log:*:*:project/Project name/savedsearch",
       "acs:log:*:*:project/Project name/savedsearch/*"
      ],
      "Effect": "Allow"
  ]
}
```

Use API operations to grant permissions to a RAM user

The permissions to write data to a specified project

To grant a RAM user only the permissions to write data to a specified project, use the following policy:

```
{
  "Version": "1",
  "Statement": [
    {
        "Action": [
            "log:Post*"
        ],
        "Resource": "acs:log:*:*:project/Project name/*",
        "Effect": "Allow"
    }
]
```

The permissions to consume data from a specified project

To grant a RAM user only the permissions to consume data from a specified project, use the following policy:

Cloud Defined Storage

٠

(
"Version": "1",
"Statement": [
(
"Action": [
"log:ListShards",
"log:GetCursorOrData",
"log:GetConsumerGroupCheckPoint",
"log:UpdateConsumerGroup",
"log:ConsumerGroupHeartBeat",
"log:ConsumerGroupUpdateCheckPoint",
"log:ListConsumerGroup",
"log:CreateConsumerGroup"
],
"Resource": "acs:log:*:*:project/Project name/*",
"Effect": "Allow"
}
]
}
The permissions to consume data from a specified Logstore
To grant a RAM user only the permissions to consume data from a specified Legisteric use the following policy:
To grant a KAM user only the permissions to consume data from a specified Logstore, use the following poincy.
{
"Version": "1",
"Statement": [
{
"Action": [

```
og:List
        "log:GetCursorOrData",
        "log:GetConsumerGroupCheckPoint",
        "log:UpdateConsumerGroup",
        "log:ConsumerGroupHeartBeat",
        "log:ConsumerGroupUpdateCheckPoint",
        "log:ListConsumerGroup",
        "log:CreateConsumerGroup"
      ],
      "Resource": [
        "acs:log:*:*:project/Project name/logstore/Logstore name",
        "acs:log:*:*:project/Project name/logstore/Logstore name/*"
      ],
      "Effect": "Allow"
    }
 ]
}
```

4.12. Monitor Log Service

4.12.1. Overview

The service log feature of Log Service helps you record log data about the operations that are performed on the resources of a project. This feature also provides dashboards that allow you to analyze data in multiple dimensions. You can use this feature to view the service status of Log Service in real time and improve O&M efficiency.

Default configurations

Default configuration item	Description	
Logstore	 When the service log feature is enabled for a project, the generated log data is classified and stored in one of the dedicated Logstores. By default, Log Service automatically creates the following dedicated Logstores: internal-operation_log: stores operation logs. A log corresponds to an API request. By default, log data in the Logstore is retained for 30 days. The billing method for the Logstore is the same as regular Logstores. internal-diagnostic_log: stores the consumption delay logs of consumer groups and Logtail heartbeat logs. The logs are classified by topic. By default, log data in the Logstore is retained for 30 days. The Logstore is free of charge. For more information about log types and fields, seeLog types. (?) Note The dedicated Logstores are used to store only the logs that are generated by Log Service. You cannot write other data to these Logstores. However, you can query, analyze, and consume the data in the dedicated Logstores and configure alert rules. 	
Region	 If you select Automatic creation (recommended), Log Service automatically creates a project in the same region to store service logs. You can also select a project from the Log Storage Location drop-down list to store service logs. The selected project must reside in the same region as the project for which the service log feature is enabled. 	
Shard	By default, the system creates two shards and enables the automatic sharding feature for each Logstore. For more information, see Manage shards in Log Service User Guide .	
Log retention period	By default, log data is retained for 30 days. You can change the retention period. For more information, see Manage a Logstore in Log Service User Guide.	
Index	By default, the indexing feature is enabled for all collected log data. If you no longer need to query and analyze data or configure alert rules, you can click Index Attributes in the upper-right corner of the Search & Analysis page to disable the indexing feature.	

Dashboard	The following dashboards are automatically created: • Operations Statistics • Logtail Collection Statistics • Logtail Monitoring • Consumer Group Monitoring For more information, see Service log dashboards.
-----------	---

Scenarios

• Check whether data is evenly written and consumed among shards

You can use predefined dashboards to view the data write and consumption trends of shards and check whether data is evenly written or consumed among shards.

Multiple Logstores in a project may share the same shards. To view the data writes to multiple shards of a Logstore, you can specify the Logstore as a filter condition.

• Monitor API request status

You can call API operations to write log data, consume log data, and create projects or Logstores. A log is generated in the **internal-operation_log** Logstore each time an API operation is called. If an API request fails, the value of the **Status** field in the generated log is an integer that is greater than 200. For example, the value of the field is 404. You can monitor API requests by viewing the number of logs in which the value of the **Status** field is greater than 200.

• View Logtail status

By default, the following Logtail-related dashboards are created after you enable the service log feature: Logtail Monitoring and Logtail Collection Statistics. The Logtail Monitoring dashboard helps you monitor Logtail exceptions such as regular expression mismatches and log data parsing failures.

4.12.2. Manage service logs

This topic describes how to enable and disable the service log feature. This topic also describes how to modify service log configurations.

Prerequisites

- A project is created. For more information, see Manage a project in Log Service User Guide.
- The RAM user that you want to use to log on to the Log Service console is granted the required permissions. You must use an Apsara Stack tenant account to grant the required permissions to a RAM user. For more information, see Grant permissions to a RAM role in Log Service User Guide.

Enable the service log feature

- 1. Log on to the Log Service console
- For more information, see Log on to the Log Service console in Log Service User Guide.
- 2. In the Projects section, click the name of the project that you want to manage.
- 3. In the Operations Log section of the Overview page, click Enable Service Logs.
- 4. In the Enable Service Logs panel, configure the parameters and click OK. The following table describes the parameters.

Parameter	Description		
Service Logs	 Detailed Logs: records logs related to operations that are performed on the resources in your project, including create, modify, delete, read, and write operations. The logs are stored in the internal-operation_log Logstore of a specified project. Important Logs: records the consumption delay logs of consumer groups and Logtail heartbeat logs by Logstore. The logs are stored in the internal-diagnostic_log Logstore of a specified project. 		
Les Charges Lessier	• Automatic creation (recommended): Log Service automatically creates a project named log-service-{User ID}- {region} in the same region to store service logs. We recommend that you store all service logs of the same region in this project.		
Log Storage Location	 Current Project: Log Service stores service logs in the current project. 		
	 Other projects in the drop-down list: Log Service stores service logs in another project that resides in the same region as the current project. You can specify only the project that resides in the same region as the project for which the service log feature is enabled. 		

Modify service log configurations

1. Log on to the Log Service console.

- For more information, see Log on to the Log Service console in Log Service User Guide.
- 2. In the Projects section, click the name of the project that you want to manage.
- 3. In the **Operations Log** section of the **Overview** page, click **Modify**.
- 4. In the Service Logs section of the Modify Service Logs Settings panel, select the log type that you want to record and clear the log type that you do not need to record.
- 5. Select a project in which you want to store service logs from the Log Storage Location drop-down list.

? Note

- We recommend that you select **Automatic creation (recommended)** to store service logs in the project that is automatically created. You can store the service logs of different projects that reside in the same region in the same project.
- After you change the value of Log Storage Location, the service logs that are generated after the change are stored in the new project that you specify. The logs that are stored in the original project are not automatically deleted or migrated to the new project that you specify. If you no longer need the logs in the original project, you can manually delete the original project.

6. Click **OK**.

Disable the service log feature

1. Log on to the Log Service console

- For more information, see Log on to the Log Service console in Log Service User Guide.
- $\ensuremath{\mathsf{2.}}$ In the Projects section, click the name of the project that you want to manage.

- 3. In the Operations Log section of the Overview page, click Modify.
- 4. In the Modify Service Logs Settings panel, clear all log types that are selected in the Service Logs section.
- 5. Click \mathbf{OK} to disable the service log feature.

() Note After you disable the service log feature, Log Service does not delete the service logs that are stored in the specified project. You can manually delete the project to delete the service logs that you no longer need.

Grant permissions to a RAM user

Before you can use the service log feature as a RAM user, you must use your Apsara Stack tenant account to grant the required permissions to the RAM user. For more information, see **Grant permissions to a RAM role** in **Log Service User Guide**. The following sample code provides an example of a policy that contains the required permissions:

```
"Version": "1",
 "Statement": [
   {
      "Action": [
       "log:CreateDashboard",
       "log:UpdateDashboard"
      1,
      "Resource": "acs:log:*:*:project/{The project in which logs are stored}/dashboard/*",
      "Effect": "Allow"
   },
    {
      "Action": [
       "log:GetProject",
        "log:CreateProject",
       "log:ListProject"
      ],
     "Resource": "acs:log:*:*:project/*",
"Effect": "Allow"
    },
      "Action": [
       "log:List*",
       "log:Create*",
       "log:Get*",
       "log:Update*"
      ],
      "Resource": "acs:log:*:*:project/{The project in which logs are stored}/logstore/*",
      "Effect": "Allow"
    },
   {
     "Action": [
       "log:*"
     1,
      "Resource": "acs:log:*:*:project/{The project for which the service log feature is enabled}/logging",
      "Effect": "Allow"
   }
 ]
}
```

4.12.3. Log types

Log Service provides the service log feature. You can use this feature to generate different types of logs. This topic describes the log types and the fields for each log type.

Log types

If you enable the service log feature, you must select the types of the logs that you want to generate. The following table describes the log types.

Log type	Overview	Logstore	Log source	Description
Detailed Logs	Records the operations that are performed on the resources in your project, including create, modify, delete, read, and write operations.	internal- operation_log	Operation logs	The logs of all API requests, including requests that are sent in the Log Service console and by using consumer groups and SDKs.
			Consumption delay logs of consumer groups	The consumption delay logs of consumer groups. These logs are generated at 2-minute intervals. If you want to query the consumption delay logs of a consumer group, you must specifytopic_: consumergroup_log in the query statement.
			Logtail alert logs	The alert logs that record errors on Logtail. Alert logs are generated at 30-second intervals. If the same error occurs multiple times within 30 seconds, only one alert log is generated. The alert log contains the total number of times that the error occurs and one error message. If you want to query Logtail alert logs, you must specify topic_: logtail_alarm in the query statement.
Important Logs	Records the consumption delay events of consumer groups and events that are related to the errors, heartbeats, and statistics of Logtail by Logstore.	internal- diagnostic_log		

Logtail collection logs	The collection logs that record statistics about Logtail configurations. These logs are generated at 10-minute intervals. If you want to query Loqtail collection logs, you must specifytopic: logtail_profilein the query statement.
Logtail status logs	The status logs of Logtail. Logtail reports status at regular intervals. These logs are generated at 1-minute intervals. If you want to query Logtail status logs, you must specifytopic: logtail_status in the query statement.

() **Important** To ensure the compatibility of a custom query statement, we recommend that you use __topic_: xxx to specify a log type in the query statement.

Operation logs

Operation logs are classified into the following categories based on the Method field: read operation logs, write operation logs, and resource operation logs. The following table describes the categories of operation logs.

Category	Request method
Read operation log	Read operation logs are generated when you call the following API operations: • GetHistograms • GetLogs • PullLogs • GetCursor • GetCursorTime
Write operation log	Write operation logs are generated when you call the following API operations:PutLogsPutWebtracking
Resource operation log	Resource operation logs are generated when you call the following API operations: API operations such as CreateProject and DeleteProject

The following table describes the common fields in operation logs.

Field	Description	Example
APIVersion	The version of the API.	0.6.0
AccessKeyId	The AccessKey ID of the account that is used to access Log Service.	LTAI4FkSqNGBsVTqVZYx****
CallerType	The type of the API caller.	Subuser
InvokerUid	The ID of the account that is used to call the API operation.	175921811532****
Latency	The latency of the request. Unit: microseconds.	123279
LogStore	The name of the Logstore.	logstore-1
Method	The API operation for which the log is recorded.	GetLogs
NetOutFlow	The volume of read traffic. Unit: bytes.	120
NetworkOut	The volume of read traffic that is received over the Internet. Unit: bytes.	10
Project	The name of the project.	project-1
RequestId	The ID of the request.	8AEADC8B0AF2FA2592C9****
SourceIP	The IP address of the client that sends the request.	1.2.3.4
Status	The HTTP status code in the response to the request.	200
UserAgent	The agent that is used by the client to call the API operation.	sls-java-sdk-v-0.6.1

The following table describes the fields that are specific to read operation logs.

Field	Description	Example
BeginTime	The start time of the request. The value is a UNIX timestamp.	1523868463
DataStatus	The response to the request. Valid values include Complete, OK, and Unknown.	ОК
EndTime	The end time of the request. The value is a UNIX timestamp.	1523869363
Offset	The read offset that you specify when you call the GetLogs operation.	20

Query	The original query statement.	UserAgent: [consumer-group-java]*
RequestLines	The number of rows that are requested by the caller.	100
ResponseLines	The number of returned rows.	100
Reverse	 Indicates whether logs are returned in descending order by timestamp. 1: Logs are returned in descending order by timestamp. 0: Logs are returned in ascending order by timestamp. This is the default value. 	0
TermUnit	The number of delimited keywords that are included in the search statement.	0
Торіс	The topic of the data that is read.	topic-1

The following table describes the fields that are specific to write operation logs.

Field	Description	Example
InFlow	The size of the raw data that you want to write. Unit: bytes.	200
InputLines	The number of lines that you want to write.	10
NetInflow	The size of the compressed data that you want to write. Unit: bytes.	100
Shard	The ID of the shard to which data is written.	1
Торіс	The topic of the data that is written.	topic-1

Consumption delay logs of consumer groups The following table describes the fields in consumption delay logs.

Field	Description	Example
consumer_group	The name of the consumer group.	consumer-group-1
fallbehind	The interval between the current consumption checkpoint and the point in time at which the last write operation log is recorded. Unit: seconds.	12345
logstore	The name of the Logstore.	logstore-1
project	The name of the project.	project-1
shard	The ID of the shard whose data is consumed.	1

Logtail alert logs

The following table describes the fields in Logtail alert logs.

Field	Description	Example
alarm_count	The number of times that alerts are generated in the specified time window.	10
alarm_message	The sample raw log that triggers the alert.	M_INFO_COL,all_status_monitor,T22380,0,2018-04-17 10:48:25.0,AY66K,AM5,2018-04-17 10:48:25.0,2018- 04-17 10:48:30.561,i- 23xebl5ni.1569395.715455,901,00789b
alarm_type	The type of the alert.	REGISTER_INOTIFY_FAIL_ALARM
logstore	The name of the Logstore.	logstore-1
os	The operating system. Example: Linux or Windows.	Linux
project	The name of the project.	project-1
source_ip	The IP address of the server on which Logtail runs.	1.2.3.4
version	The version of Logtail.	0.14.2

Logtail collection logs

 $\label{eq:logical} \mbox{Logitail collection logs are classified into the following categories based on the $file_name$ field:$

• Statistics about a Logtail configuration for a log file.

The following table describes the fields in Logtail collection logs.

Field	Description	Example
logstore	The name of the Logstore.	logstore-1

config_name	The name of the Logtail configuration. The name is alobally unique and must be in the following format: ##Logtail configuration version##Project name\$Configuration name .	##1.0##project-1\$logstore-1
error_line	The raw log that causes an error.	M INFO_COL,all_status_monitor,T22380,0,2018-04-17 10:48:25.0,AY66K,AM5,2018-04-17 10:48:25.0,2018- 04-17 10:48:30.561,i- 23xebl5ni.1569395.715455,901,00789b
file_dev	The device ID of the log file.	123
file_inode	The inode of the log file. Note If the file_name field is set to logstore_statistics , this field is invalid.	124
file_name	The full path of the log file or the value of logstore_statistics .	/abc/file_1
file_size	The size of the log file. Unit: bytes.	12345
history_data_failures	The number of times that data fails to be processed.	0
last_read_time	The last read time in the specified time window. The value is a UNIX timestamp.	1525346677
project	The name of the project.	project-1
logtail_version	The version of Logtail.	0.14.2
os	The operating system.	Windows
parse_failures	The number of lines that fail to be parsed in the specified time window.	12
read_avg_delay	The average of the difference between the actual file size and the offset value that is generated each time log data is read in the specified time window.	65
read_count	The number of reads in the specified time window.	10
read_offset	The last read offset of the log file. Unit: bytes.	12345
regex_match_failures	The number of times that regular expressions fail to be matched.	1
send_failures	The number of times that logs fail to be sent in the specified time window.	12
source_ip	The IP address of the server on which Logtail runs.	1.2.3.4
succeed_lines	The number of log lines that are processed.	123
time_format_failures	The number of times that log times fail to be matched.	122
total_bytes	The total size of data that is read. Unit: bytes.	12345

 $The following table describes the fields that are specific to Logstore statistics collected when the {fielg_name} field is set to {logstore_statistics}.$

Field	Description	Example
send_block_flag	Indicates whether the send queue is blocked when the specified time window ends.	false
send_discard_error	The number of packets that are discarded due to data errors or insufficient permissions in the specified time window.	0
send_network_error	The number of packets that fail to be sent due to network errors in the specified time window.	12
send_queue_size	The number of unsent packets in the current send queue when the specified time window ends.	3
send_quota_error	The number of packets that fail to be sent because the Logtail quota is exceeded in the specified time window.	0
send_success_count	The number of packets that are sent in the specified time window.	12345
sender_valid_flag	 Indicates whether the send flag of the current Logstore is valid when the specified time window ends. Valid values: true: The flag is valid. false: The flag is disabled due to network or quota errors. 	true

max_send_success_time	The last time when data was sent in the specified time window. The value is a UNIX timestamp.	1525342763
max_unsend_time	The last time when packets in the send queue failed to be sent in the specified time window. The value is a UNIX timestamp. If the send queue is empty, the value is 0.	1525342764
min_unsend_time	The first time when packets in the send queue failed to be sent in the specified time window. The value is a UNIX timestamp. If the send queue is empty, the value is 0.	1525342764

Logtail status logs

The following table describes the fields in Logtail status logs.

Field	Description	Example
сри	The CPU load of the Logtail process.	0.001333156
hostname	The hostname.	abc2.****
instance_id	The ID of the instance. This ID is randomly assigned.	05AFE618-0701-11E8-A95B- 00163E025256_10.11.12.13_151745****
ip	The IP address.	1.0.1.0
load	The average system load.	0.01 0.04 0.05 2/376 5277
memory	The memory space that is occupied by the Logtail process. Unit: MB.	12
detail_metric	The metrics in the JSON format.	detail_metric
os	The operating system.	Linux
os_cpu	The CPU utilization of the system.	0.004120005
os_detail	The details of the operating system.	2.6.32-220.23.8.tcp1.34.el6.x86_64
status	The status of Logtail. • ok • busy • many_log_files • process_block • send_block • send_error	busy
user	The username.	root
user_defined_id	The user-defined ID.	aliyun-log-id
uuid	The universally unique identifier (UUID) of the server.	64F28D10-D100-492C-8FDC-0C62907F****
version	The version of Logtail.	0.14.2
project	The project to which the Logtail configuration belongs.	my-project

The following table describes the fields that are included in the detail_metric field.

Field	Description	Example
config_count	The number of Logtail configurations.	1
config_get_last_time	The last time when the Logtail configuration was obtained.	2021-07-20 16:19:22
config_update_count	The number of Logtail configuration updates after Logtail was started.	1
config_update_item_count	The total number of configuration items that are updated after Logtail was started.	1
config_update_last_time	The time when the configuration was last updated after Logtail was started.	2021-07-20 16:18:42
env_config	Indicates whether environment variables are used to create the Logtail configuration.	false
event_tps	The transactions per second (TPS).	1
last_read_event_time	The last time when data was read.	2021-07-20 16:18:42
last_send_time	The last time when data was sent.	2021-07-20 16:18:42
multi_config	Indicates whether multiple Logtail configurations are enabled to collect logs from the same file.	false
net_err_stat	The number of times that network sending errors occurred in the previous 1, 5, and 15 minutes.	0,0,0
open_fd	The number of log files that are open.	1

User Guide-Log Service

Indicates whether Logtail plug-ins are enabled.	false
The number of monitored log files that are modified.	1
The number of scanned directories.	1
The number of scanned files.	1
The size of log data that is processed per second. Unit: bytes.	1000
The number of logs that are processed per second.	1000
The number of processing queues that reach the maximum processing capacity.	1
The total number of processing queues.	10
The number of data processing transactions per second.	0
The number of log files that are being processed.	1
The region where Logtail resides.	cn-hangzhou,cn-shanghai
The number of directories to be monitored.	1
The size of raw log data that is sent per second. Unit: bytes.	11111
The number of logs that are sent per second.	1000
The volume of network data that is sent per second. Unit: bytes.	1000
The number of send queues that reach the maximum sending capacity.	1
The total number of send queues.	12
The maximum number of packets that can be concurrently sent from send queues.	10
The number of data sending transactions per second.	0.075
The number of abnormal send queues.	0
The start time.	2021-07-20 16:19:22
The number of packets that are concurrently sent.	0
	Indicates whether Logtail plug-ins are enabled.The number of monitored log files that are modified.The number of scanned directories.The number of scanned files.The size of log data that is processed per second.Unit: bytes.The number of processing queues that reach the maximum processing capacity.The total number of processing queues.The number of data processing transactions per second.The number of directories to be monitored.The region where Logtail resides.The number of logs that are sent per second.The number of send queues that reach the maximum sending capacity.The total number of packets that can be concurrently sent from send queues.The start time.The number of abormal send queues.

4.12.4. Service log dashboards

After you enable the service log feature, Log Service automatically creates the following dashboards based on the log types that you selected to display the related statistics: Operations Statistics, Logtail Collection Statistics, Logtail Monitoring, and Consumer Group Monitoring.

Dashboards

- When you enable the service log feature for a project, you can select one or more of the following log types:
- If you turn on Detailed Logs, Log Service automatically creates the Operations Statistics dashboard. For more information, see Operations Statistics.
- If you turn on Important Logs, Log Service automatically creates the following dashboards: Logtail Collection Statistics, Logtail Monitoring, and Consumer Group Monitoring. For more information, see Logtail Collection Statistics, Logtail Monitoring, and Consumer Group Monitoring.

Operations Statistics

This dashboard displays the statistics about visits and operations, such as API requests and operations that are performed on projects.

Logtail Collection Statistics

This dashboard displays the statistics about the logs collected by Logtail.

Logtail Monitoring

This dashboard displays the statistics about Logtail errors and alerts to help you monitor the status of Logtail in real time.

Consumer Group Monitoring

This dashboard displays the statistics about consumer groups, including the volume of data consumed from shards, consumption delay, and the details of consumer groups.

4.13. FAQ

4.13.1. Log collection

4.13.1.1. How do I troubleshoot errors that occur when I use Logtail to collect

logs?

If the preview page is blank or the No Data message appears on the query page after you create a Logtail configuration to collect logs, perform the steps that are described in the topic to troubleshoot the issue.

Procedure

1. Check whether Log Service receives heartbeats from the machine group.

You can view the Logtail heartbeat status in the Log Service console. For more information, see View the status of a server group.

If the heartbeat status is OK, perform the next step. If the heartbeat status is FAIL, identify the cause of the failure. For more information, see What can I do if Log Service does not receive heartbeats from a Logtail client?.

- Check whether the Logtail configuration is created. If the heartbeat status of Logtail is OK, check whether the Logtail configuration is created. Make sure that the path and name of monitored logs match those of the files stored on the server. The path can be a full path or a path that includes wildcards.
- Make sure that the Logtail configuration is applied to the machine group. For more information, see Manage server group configurations.
- 4. Check collection errors.

If the Logtail configuration is valid, check whether new logs are generated in real time. Logtail collects only incremental log data. Logtail does not read log files that are not updated. If a log file is updated but the updated data cannot be found in Log Service, you can use the following method to troubleshoot the issue:

• View the logs of the Logtail client.

Client logs include key INFO logs, all WARNING logs, and all ERROR logs. To view complete and real-time error information, view the client logs in the following paths:

- Linux: /usr/local/ilogtail/ilogtail.LOG.
- Linux: /usr/local/ilogtail/logtail_plugin.LOG. The file contains the logs such as HTTP logs, MySQL binary logs, and MySQL query results.
- Windows x64 : C:\Program Files (x86)\Alibaba\Logtail\logtail_*.log
- Windows x32 : C:\Program Files\Alibaba\Logtail\logtail_*.log
- Check whether the amount of log data exceeds the limit.

To collect large amounts of log data, you may need to modify the startup parameters of Logtail to increase the log collection throughput. For more information, see Configure the startup parameters of Logtail.

4.13.1.2. What can I do if Log Service does not receive heartbeats from a Logtail

client?

If Log Service does not receive heartbeats from a Logtail client, perform the steps that are described in the topic to troubleshoot the issue.

Background information

After Logtail is installed on a server, the Logtail client sends heartbeats to Log Service. If the status page of the machine group shows that Log Service does not receive heartbeats from a Logtail client, the Logtail client is not installed or disconnected from the server.

Step 1: Check whether Logtail is installed

Use the following method to check whether Logtail is installed:

• On a Linux server, run the following command:

sudo /etc/init.d/ilogtaild status

If Logtail is installed, the following result appears:

ilogtail is running

- · On a Windows server, perform the following steps:
- i. On Control Panel, click Administrative Tools, and then click Services.

ii. In the Services window, check the status of the LogtailDaemon and LogtailWorker services. If the services are in the Running state, Logtail is installed.

If Logtail is not installed, install Logtail. For more information, see Install Logtail on a Linux server or Install Logtail on a Windows server. Make sure that you install Logtail based on the region where your Log Service project resides. If Logtail is running, go to the next step.

Step 2: Check the Log Service endpoint in the Logtail installation command

When you install Logtail, you must specify a Log Service endpoint based on the region where your Log Service project resides. If the endpoint is incorrect or the Logtail installation command is invalid, Log Service cannot receive heartbeats from the Logtail client. You can view the Log Service endpoint and the installation method in the Logtail configuration file named ilogtail config.json. The file is stored in the

- Linux: /usr/local/ilogtail/ilogtail config.json
- 64-bit Windows: C:\Program Files (x86)\Alibaba\Logtail\ilogtail_config.json
- 32-bit Windows: C:\Program Files\Alibaba\Logtail\ilogtail config.json

In the ilogtail_config_ison Logtail configuration file, check the endpoint that is specified for the config_server_address parameter. Then, check
whether the Logtail client can use the endpoint to connect to Log Service. For example, if the endpoint that is recorded in the ilogtail_config.json Logtail
configuration file is logtail.cn-qingdao-env25-d01.sls-pub.inter.env25.shuguang.com, you can run the following command to check the connection:

Linux:

curl logtail.cn-qingdao-env25-d01.sls-pub.inter.env25.shuguang.com

• Windows:

following path:

telnet logtail.cn-qingdao-env25-d01.sls-pub.inter.env25.shuguang.com 80

If the Log Service endpoint in the Logtail installation command is incorrect, re-install Logtail. For more information, see Install Logtail on a Linux server or Install Logtail on a Windows server.

If the Log Service endpoint in the Logtail installation command is correct, go to the next step.

Step 3: Check the server IP addresses in the machine group

The server IP address that is obtained by a Logtail client must be configured in the machine group. Otherwise, Log Service cannot receive heartbeats or collect logs from the Logtail client. Logtail uses the following methods to obtain the IP address of a server:

• If the server is not bound with a hostname, Logtail obtains the IP address of the first network interface controller (NIC) card of the server.

• If the server is bound with a hostname, Logtail obtains the IP address that corresponds to the hostname. You can view the hostname and IP address in the /etc/hosts file.

? Note

You can run the hostname command to view the hostname.

Perform the following steps to check whether the server IP address that is obtained by the Logtail client is configured in the machine group.

- 1. Check the server IP address that is obtained by Logtail.
 - The value of the ip field in the app_info.json file is the server IP address that is obtained by Logtail. The file is stored in the following path:
 - Linux: /usr/local/ilogtail/app_info.json
 - 64-bit Windows: C:\Program Files (x86)\Alibaba\Logtail\app_info.json
 - 32-bit Windows: C:\Program Files\Alibaba\Logtail\app_info.json

? Note

• If the ip field in the app_info.json file is empty, Logtail cannot work. In this case, you must configure an IP address for the server and restart Logtail.

- The app_info.json file is used only to record information. If you modify the IP address in the file, the server IP address that is obtained by Logtail is not updated.
- 2. Check the server IP addresses in the machine group.

Log on to the Log Service console. In the Projects section, click the project to which the machine group belongs. In the left-side navigation pane, choose **Resources > Machine Groups**, and then click the name of the machine group. In the Machine Group Status section of the **Machine Group Settings** page, check the server IP addresses.

If no server IP address in the machine group is the same as the IP address that is obtained by Logtail, perform the following step to modify the IP address configurations in the Log Service console:

- If a server IP address in the machine group is incorrect, change the IP address to the IP address that is obtained by Logtail. Then, check the heartbeat status 1 minute after you save the change.
- If you modify the IP address of the server where Logtail is installed, for example, the/etc/hosts file, restart Logtail. After Logtail obtains the new server IP address, set a server IP address in the machine group to the value of the __ip_field in the __app_info.json__file.

You can use the following method to restart Logtail:

• On a Linux server, run the following commands:

sudo /etc/init.d/ilogtaild stop
sudo /etc/init.d/ilogtaild start

- On a Windows server, perform the following steps:
- On Control Panel, choose Administrative Tools > Services. In the list that appears, find LogtailWorker and restart LogtailWorker.

4.13.1.3. How do I query the status of local log collection?

You can use the status query feature of Logtail to query the health status of Logtail and the log collection status. You can also use this feature to troubleshoot log collection issues and customize status monitoring for log collection.

Usage notes

After you install a Logtail client that supports the status query feature, you can query the local log collection status by running commands on the client. For more information about how to install Logtail, see Install Logtail on a Linux server.

You can run the /etc/init.d/ilogtaild -h command on a client to check whether the client supports the status query feature. If the result includes the logtail insight, version keyword, the client supports the status query feature.

```
/etc/init.d/ilogtaild -h
Usage: ./ilogtaild { start | stop (graceful, flush data and save checkpoints) | force-stop | status | -h for help}$
logtail insight, version : 0.1.0
commond list :
    status all [index]
        get logtail running status
        status attive [--logstore | --logfile] index [project] [logstore]
        list all active logstore | logfile. if use --logfile, please add project and logstore. default --logstore
        status logstore [--format=line | json] index project logstore
        get logstore status with line or json style. default --format=line
        status logfile [--format=line | json] index project logstore fileFullPath
        get log file status with line or json style. default --format=line
        status history beginIndex endIndex project logstore [fileFullPath]
        query logstore | logfile history status.
index : from 1 to 60. in all, it means last $(index) minutes; in active/logstore/logfile/history, it means last $(index)*10 minutes)
}
```

Logtail supports multiple query commands. The following table describes the query commands, command functionalities, time ranges, and time windows for query results.

Command	Functionality	Maximum time range that can be queried	Time window
all	Queries the status of Logtail.	Last 60 minutes	1 minute
active	Queries the active Logstores that are collecting logs and the active log files from which logs are being collected.	Last 600 minutes	10 minutes

logstore	Queries the collection status of a Logstore.	Last 600 minutes	10 minutes
logfile	Queries the collection status of a log file.	Last 600 minutes	10 minutes
history	Queries the collection status of a Logstore or log file within a specified period of time.	Last 600 minutes	10 minutes

? Note

- The index parameter in the preceding commands specifies the index of the time window. Valid values: 1 to 60. The index of the latest time window is 1. The time window ends at the current system time. If you specify a 1-minute time window, the status in the previous interval of (index, index-1) minutes is returned. If you specify a 10-minute time window, the status in the previous interval of (10*index, 10*(index-1)) minutes is returned.
- All commands in the preceding table are subcommands of the status command.

all command

• Syntax

/etc/init.d/ilogtaild status all [index]

? Note

The all command is used to query the status of Logtail. The index parameter is optional. Default value: 1.

• Examples

/etc/init.d/ilogtaild status all 1 ok

/etc/init.d/ilogtaild status all 10 busy

Response

Status	Description	Priority	Troubleshooting
ok	Logtail runs as expected.	N/A	No action is required.
busy	The collection speed is high, and Logtail runs as expected.	N/A	No action is required.
many_log_files	A large number of log files are being collected by Logtail.	Low	You can check whether Logtail is configured to collect log files that do not need to be collected.
process_block	Log parsing is blocked.	Low	You can check whether a large number of logs are generated in a short period of time. If you use the all command multiple times and the returned value is always process_block, you can modify the limit of CPU utilization or the limit of concurrent packet sending. For more information, see Configure the startup parameters of Logtail.
send_block	The process of packet sending is blocked.	High	You can check whether a large number of logs are generated in a short period of time and whether the network is stable. If you use the all command multiple times and the returned value is always send_block, you can modify the limit of CPU utilization or the limit of concurrent packet sending. For more information, see Configure the startup parameters of Logtail

active command

• Syntax

O Note The active command is used to query log files. We recommend that you query active Logstores before you query the active log files in the Logstores.

/etc/init.d/ilogtaild status active [--logstore] index

You can use the active [--logstore] index command to query all active Logstores. The --logstore parameter is optional.

/etc/init.d/ilogtaild status active --logfile index project-name logstore-name

You can use the active --logfile index project-name logstore-name command to query all active log files in the Logstore of a project.

Examples

/etc/init.d/ilogtaild status active 1
sls-zc-test : release-test
sls-zc-test : release-test-ant-rpc-3
sls-zc-test : release-test-same-regex-3

If you run the active --logstore index command, the names of the active Logstores are returned in the following format: project-name :

/etc/init.d/ilogtaild status active --logfile 1 sls-zc-test release-test /disk2/test/normal/access.log

• If you run the active --logfile index project-name logstore-name command, the full paths of active log files are returned.

• The status of inactive Logstores or inactive log files in the query time window is not returned.

logstore command

Syntax

/etc/init.d/ilogtaild status logstore [--format={line|json}] index project-name logstore-name

? Note

- The logstore command is used to query the collection status of the specified project and Logstore in the Line or JSON format.
- The default value of the --formate parameter is --formateline . This value indicates that the status is returned in the LINE format.
- If the Logstore specified in the preceding command does not exist or is not active in the query time window, an empty response in the LINE format or the null value in the JSON format is returned.

Examples

/etc/init.d/ilogtaild status logstore 1 sls-zc-test release-test-same time_begin_readable : 17-08-29 10:56:11 time_end_readable : 17-08-29 11:06:11 time_begin : 1503975371 time_end : 1503975971 project : sls-zc-test logstore : release-test-same status : ok config : ##1.0##sls-zc-test\$same read_bytes : 65033430 parse_success_lines : 230615 parse fail lines : 0 last_read_time : 1503975970 read_count : 687 avg_delay_bytes : 0 max_unsend_time : 0 min unsend time : 0 max_send_success_time : 1503975968 send_queue_size : 0 send_network_error_count : 0 send_network_quota_count : 0 send_network_discard_count : 0 send_success_count : 302 send_block_flag : false sender_valid_flag : true /etc/init.d/ilogtaild status logstore --format=json 1 sls-zc-test release-test-same "avg_delay_bytes" : 0, "config" : "##1.0##sls-zc-test\$same", "last_read_time" : 1503975970, "logstore" : "release-test-same" "max_send_success_time" : 1503975968, "max_unsend_time" : 0, "min_unsend_time" : 0, "parse_fail_lines" : 0, "parse_success_lines" : 230615,
"project" : "sls-zc-test", "read bytes" : 65033430, "read_count" : 687, "send_block_flag" : false, "send_network_discard_count" : 0, "send_network_error_count" : 0, "send_network_quota_count" : 0, "send_queue_size" : 0, "send_success_count" : 302, "sender_valid_flag" : true, "status" : "ok", "time begin" : 1503975371, "time begin readable" : "17-08-29 10:56:11", "time_end" : 1503975971, "time_end_readable" : "17-08-29 11:06:11" 1

Response

Parameter	Description	Unit
status	The status of the Logstore. For information about the different status of Logstore and the actions that are required to handle each status, see the following table.	N/A
time_begin_readable	The time when logs become readable.	N/A
time_end_readable	The time when logs become unreadable.	N/A
time_begin	The time when statistics collection starts.	UNIX timestamp in seconds
time_end	The time when statistics collection ends.	UNIX timestamp in seconds
project	The name of the project.	N/A
logstore	The name of the Logstore.	N/A
config	The name of the Logtail configuration. The name is globally unique. The format of the name is ##1.0## + project + \$ + config.	N/A
read_bytes	The amount of log data that is read in the query time window.	Bytes
parse_success_lines	The number of log lines that are parsed in the query time window.	Lines
parse_fail_lines	The number of log lines that fail to be parsed in the query time window.	Lines
last_read_time	The time when logs are last read in the query time window.	UNIX timestamp in seconds
read_count	The number of times that the log file is read in the query time window.	N/A
avg_delay_bytes	The average of the difference between the actual file size and the offset value that is generated each time log data is read in the query time window.	Bytes
max_unsend_time	The maximum waiting period for an unsent packet in the sending queue. An unsent packet refers to a packet that is still in the sending queue at the end of the query time window. If no packets exist in the queue, the value is 0.	UNIX timestamp in seconds
min_unsend_time	The maximum waiting period for an unsent packet in the sending queue. An unsent packet refers to a packet that is still in the sending queue at the end of the query time window. If no packets exist in the queue, the value is 0.	UNIX timestamp in seconds
max_send_success_time	The maximum waiting period for an unsent packet in the sending queue.	UNIX timestamp in seconds
send_queue_size	The number of unsent packets in the sending queue at the end of the query time window.	N/A
send_network_error_count	The number of packets that cannot be sent due to network errors in the query time window.	N/A
send_network_quota_count	The number of packets that cannot be sent due to quota limit in the query time window.	N/A
send_network_discard_count	The number of packets that are discarded due to data errors or lack of permissions.	N/A
send_success_count	The number of packets that are sent in the query time window.	N/A

send_block_flag	Indicates whether the sending queue is blocked at the end of the query time window.	N/A
sender_valid_flag	Indicates whether the sender flag of the Logstore is valid. The value true indicates that the sender flag is valid. The value false indicates that the sender flag is invalid and disabled due to a network error or quota error.	N/A

Logstore status

Status	Description	Troubleshooting
ok	Logtail runs as expected.	No action is required.
process_block	Log parsing is blocked.	You can check whether a large number of logs are generated in a short period of time. If you use the all command multiple times and the returned value is always process block, you can modify the limit of CPU utilization or the limit of concurrent packet sending. For more information, see Configure the startup parameters of Logtail.
parse_fail	Logtail fails to parse logs.	You can check whether the format of logs is the same as the format that you specify in the Logtail configuration.
send_block	The process of packet sending is blocked.	You can check whether a large number of logs are generated in a short period time and whether the network is stable. If you use the all command multiple times and the returned value is always send_block, you can modify the limit of CPU utilization or the limit of concurrent packet sending. For more information, see Configure the startup parameters of Logtail

logfile command

• Syntax

/etc/init.d/ilogtaild status logfile [--format={line|json}] index project-name logstore-name fileFullPath

? Note

- The logfile command is used to query the collection status of the specified log files in the LINE or JSON format.
- The default value of the --format= parameter is --format=line . This value indicates that the status is returned in the LINE format.
- If the log file specified in the command does not exist or is not active in the query time window, an empty response in the LINE format or the null value in the JSON format is returned.
- The --format parameter is placed after the logfile parameter.
- $\circ~$ The value of the ~ $_{\tt filefullpath}~$ parameter must be set to the full path of the log file.

• Examples

/etc/init.d/ilogtaild status logfile 1 sls-zc-test release-test-same /disk2/test/normal/access.log time_begin_readable : 17-08-29 11:16:11 time_end_readable : 17-08-29 11:26:11 time_begin : 1503976571 time_end : 1503977171 project : sls-zc-test logstore : release-test-same status : ok config : ##1.0##sls-zc-test\$same file_path : /disk2/test/normal/access.log file_dev : 64800 file_inode : 22544456 file_size_bytes : 17154060 file_offset_bytes : 17154060 read_bytes : 65033430 parse_success_lines : 230615
parse_fail_lines : 0 _______last_read_time : 1503977170 read_count : 667 avg_delay_bytes : 0 /etc/init.d/ilogtaild status logfile --format=json 1 sls-zc-test release-test-same /disk2/test/normal/access.log { "avg_delay_bytes" : 0, "avg_delay_oytes" : 0, "config" : "##1.0##sls-zc-test\$same", "file_dev" : 64800, "file_inode" : 22544456, "file_path" : "/disk2/test/normal/access.log", "file_size_bytes" : 17154060, "last_read_time" : 1503977170, "logstore" : "release-test-same", "parse_fail_lines" : 0, parse_tall_ines : 0, 0, "parse_success_lines" : 230615, "project" : "sls=zc-test", "read_bytes" : 65033430, "read_count" : 667, "read_offset_bytes" : 17154060, "status" : "ok", "time_begin" : 1503976571, "time_begin_readable" : "17-08-29 11:16:11", "time_end" : 1503977171, "time_end_readable" : "17-08-29 11:26:11" 3

Response

Parameter	Description	Unit
status	The collection status of the log file in the query time window. For more information, see the status parameter in the logstore command section.	N/A
time_begin_readable	The time when logs become readable.	N/A
time_end_readable	The time when logs become unreadable.	N/A
time_begin	The time when statistics collection starts.	UNIX timestamp in seconds
time_end	The time when statistics collection ends.	UNIX timestamp in seconds
project	The name of the project.	N/A
logstore	The name of the Logstore.	N/A
file_path	The path of the log file.	N/A
file_dev	The ID of the device from which the log file is collected.	N/A
file_inode	The inode of the log file.	N/A
file_size_bytes	The size of the last log file that is scanned in the query time window.	Bytes
read_offset_bytes	The parsing offset of the log file.	Bytes
config	The name of the Logtail configuration. The name is globally unique. The format of the name is $\#\pm 1.0\#\#$ + project + $\$$ + config.	N/A
---------------------	--	---------------------------
read_bytes	The amount of log data that is read in the query time window.	Bytes
parse_success_lines	The number of log lines that are parsed in the query time window.	Lines
parse_fail_lines	The number of log lines that fail to be parsed in the query time window.	Lines
last_read_time	The time when logs are last read in the query time window.	UNIX timestamp in seconds
read_count	The number of times that the log file is read in the query time window.	N/A
avg_delay_bytes	The average of the difference between the actual file size and the offset value that is generated each time log data is read in the query time window.	Bytes

history command

• Syntax

/etc/init.d/ilogtaild status history beginIndex endIndex project-name logstore-name [fileFullPath]

? Note

- The history command is used to query the collection status of a Logstore or log file in the query time window.
- The beginIndex and endIndex parameters specify the start and end indexes of the time windows that you want to query. You must ensure that beginIndex must be less than or equal to endIndex (beginIndex <= endIndex).
- The fileFullPath parameter is optional. If you specify the path of a log file, the collection status of the log file is queried. Otherwise, the collection status of the Logstore is queried.

Examples

Query the collection status of a Logstore.

• Command

/etc/init.d/ilogtaild status history 1 3 sls-zc-test release-test-same /disk2/test/normal/access.log

Response

be	gin_time	status	read	parse_success	parse_fail	last_read_time	read_count	avg_delay	device	inode	file
_size rea 17-08-29	d_offset 11:26:11	ok	62.12MB	231000	0	17-08-29 11:36:11	671	0B	64800	22544459	18
22MB	18.22MB										
17-08-29 36MB	11:16:11 16.36MB	ok	62.02MB	230615	0	17-08-29 11:26:10	667	0B	64800	22544456	16
17-08-29	11:06:11	ok	62.12MB	231000	0	17-08-29 11:16:11	687	0B	64800	22544452	14
46MB	14.46MB										

Query the collection status of a log file.

Command

\$/etc/init.d/ilogtaild status history 2 5 sls-zc-test release-test-same

• Response

be	egin_time	status	read par	se_success parse	e_fail last_rea	d_time read_count	avg_delay send_	queue
network_er	rror quota_error	discard	error send_s	uccess send_blo	ck send_valid	max_unsend	min_unsend	max_send_success
17-08-29	11:16:11	ok	62.02MB	230615	0 17-08-29 13	:26:10 667	0B	0
0	0	300	false	true 70-01-01	08:00:00 70-01-0	08:00:00 17-08-29	11:26:08	
17-08-29	11:06:11	ok	62.12MB	231000	0 17-08-29 13	:16:11 687	0B	0
0	0	303	false	true 70-01-01	08:00:00 70-01-0	08:00:00 17-08-29	11:16:10	
17-08-29	10:56:11	ok	62.02MB	230615	0 17-08-29 13	:06:10 687	0B	0
0	0	302	false	true 70-01-01	08:00:00 70-01-0	08:00:00 17-08-29	11:06:08	
17-08-29	10:46:11	ok	62.12MB	231000	0 17-08-29 10	:56:11 692	0B	0
0	0	302	false	true 70-01-01	08:00:00 70-01-03	08:00:00 17-08-29	10:56:10	

Response

- $\circ~$ The collection status of the Logstore or log file in each query time window is listed in a line.
- $\circ \ \ \mbox{For more information about the response parameters, see the $logstore$ command and $logfile$ command sections. }$

Response status codes

Success code

If the parameters that you specify in a command is valid even if the queried Logstore or log file is not found, the code 0 is returned. Examples:

/etc/init.d/ilogtaild status logfile --format=json 1 error-project error-logstore /no/this/file
null
echo \$?
0
/etc/init.d/ilogtaild status all
ok
echo \$?
0

Error codes

/etc/init.d/ilogtaild status nothiscmd invalid param, use -h for help. echo \$? 10 /etc/init.d/ilogtaild status/all 99 invalid query interval echo \$? 1

If a non-zero code is returned, an error occurs. The following table describes the possible non-zero codes.

Code	Description	Response	Troubleshooting
10	The command is invalid or the required parameters in the command are not specified.	invalid param, use -h for help.	You can run the -h command to obtain help information.
1	The value of the index parameter is not within the range of 1 to 60.	invalid query interval	You can run the -h command to obtain help information.
1	The collection status in the specified query time window cannot be queried.	<pre>query fail, error: \$(error) . For more information, see errno.</pre>	The startup time of Logtail is earlier than the query time window. For more information, submit a ticket.
1	The start time of the query falls out of the query time window.	no match time interval, please check logtail status	You can check whether Logtail runs as expected. For more information, submit a ticket.
1	No data exists in the query time window that you specify.	invalid profile, maybe logtail restart	You can check whether Logtail runs as expected. For more information, submit a ticket.

Scenarios

You can use the status query feature of Logtail to query the overall status of Logtail. You can also obtain specific metrics based on the collection status during log collection. You can customize a mechanism to monitor the log collection status based on the queried information.

Monitor the status of Logtail

You can monitor the status of Logtail by using the all command.

For example, you can run the command every minute to query the status of Logtail. If the process_block , send_block , or send_error value is returned for 5 consecutive minutes, an alert is triggered.

You can adjust the alert duration and monitoring scope based on the priorities of the collected log files.

Monitor the log collection status

You can monitor the log collection status of a Logstore by using the <code>logstore</code> command.

For example, you can run the logstore command every 10 minutes to query the status of the Logstore. If the value of the avg_delay_bytes parameter exceeds 1 MB (1024 × 1024 bytes) or the value of the status parameter is not ok , an alert is triggered.

You can adjust the alert threshold for the avg_delay_bytes metric based on the size of data that is generated during log collection.

Check whether Logtail has finished collecting log files

You can check whether Logtail has finished collecting log files by using the logfile command.

If Logtail no longer collects log files, you can run the logfile command every 10 minutes to query the status of the log file. If the value of the file_size_bytes parameter is the same as the value of the file_size_bytes parameter, the log file is collected.

Troubleshoot log collection issues

If latency occurs on a server during log collection, you can use the history command to query the status history of log collection.

- 1. The value of the send_block_flag parameter is true. This indicates that log collection is blocked due to unstable network connections.
- If the value of the send_network_quota_count parameter is greater than 0, split the shards in the Logstore. For more information, see Split a shard.
- If the value of the send_network_error_count parameter is greater than 0, check the network connections.
- If no network error occurs, adjust the limit of concurrent packet sending and the data transfer speed of Logtail. For more information, see Configure
 the startup parameters of Logtail.

2. The parameters for packet sending are set to appropriate values. However, the value of the avg_delay_bytes parameter is large.

• Use the value of the read_bytes parameter to calculate the average speed at which logs are parsed. You can determine whether a large amount of data is transferred during log collection based on the average speed.

• Adjust the limits on resource usage for Logtail. For more information, see Configure the startup parameters of Logtail.

3. The value of the parse_fail_lines parameter is greater than 0.

Check whether the regular expression for log parsing can match all required log fields.

4.13.1.4. How do I debug a regular expression?

If you select the full regex mode when you configure Logtail to collect and parse text logs, you must specify a regular expression based on your sample log entries. This topic describes how to debug a regular expression.

Background information

To debug the regular expression that you specified in the Log Service console, you can click Validate in the console and check the following results:

- If the regular expression is used to match the start part of the first line in a log entry, check whether the regular expression can match the expected number of log entries.
- If the fields are extracted by the regular expression, check whether the value of each field meets your requirements.

You can use online tools such as regex101.com and regextester.com to debug a regular expression. You can copy and paste the regular expression that is generated by Log Service to an online tool, and specify a sample log entry as the test string.

If you use the full regex mode, Log Service automatically generates a regular expression based on the sample log entry that you specify. However, the regular expression may fail to match the message field in multi-line log entries as expected. The following example shows how to use the regex101.com tool to debug a regular expression.

Procedure

1. Log on to the Log Service console

- 2. In the Projects section, click the project that you want to manage.
- 3. On the Log Storage > Logstores tab, find the Logstore that you want to manage and choose Data Import > Logtail Configurations.
- 4. Click the name of the Logtail configuration that you want to manage.
- 5. On the Logtail Config page, copy the regular expression that is automatically generated by Log Service based on sample log entries.
- 6. Visit the regex101.com website.
- 7. In the **REGULAR EXPRESSION** field, paste the regular expression.
- On the right side of the page, you can view the explanation of the regular expression.
- 8. In the **TEST STRING** field, paste a sample log entry.

In the following figure, the log content that is included in the message field is highlighted in orange, and the log content that is not included is highlighted in blue. The figure shows that the substring that follows the at word is not included in the message field. Therefore, the regular expression does not match fields in the sample log entry as expected and cannot be used to collect log data.

REGULAR EXPRESSION	1 match, 32 steps (~1ms)
<pre>!/ \[[[^]]+)]\s\[[\w+)]\s([^:]+:\s\w+\s\w+\s[^:]+:\S+\s[^:]+:\S+\s\S+).</pre>	* / gm 🍽
TEST STRING SV	WITCH TO UNIT TESTS >
<pre>[2018-10-01T10:30:01,000] [INFO] java.lang.Exception: exception happene at TestPrintStackTrace.f(TestPrintStackTrace.java:3) at TestPrintStackTrace.g(TestPrintStackTrace.java:7) at TestPrintStackTrace.main(TestPrintStackTrace.java:16) </pre>	d

9. Check whether the regular expression can match fields in the sample log entry that contains two colons (::). The following figure shows that the regular expression fails to match fields in the sample log entry.

REGULAR EXPRESSION V1 V	no matc	n, 33 steps (~1ms)
:/\[([^]]+)]\s\[(\w+)]\s([^:]+:\s\w+\s\w+\s[^:]+:\S+\s[^:]+:\S+\s[S+ <mark>)</mark> .*	/ gm 🍽
TEST STRING	SWITCH T	O UNIT TESTS →
<pre>[2018-10-01T10:30:01,000] [INFO] java.lang.Exception: exception hap at TestPrintStackTrace.f(TestPrintStackTrace.java:3) at TestPrintStackTrace.g(TestPrintStackTrace.java7)</pre>	pened	

0. Replace the last subexpression in the regular expression with [\s\s]+, and check whether the regular expression can match fields in the sample log entries as expected.

The following figure shows how the modified regular expression matches the substring that follows the at word.

REGULAR EXPRESSION V1 V	1 match, 17 steps (~0ms)
<pre>!/ \[([^]]+)]\s\[(\w+)]\s([\S\s]+).*</pre>	/ gm 🍽
TEST STRING	SWITCH TO UNIT TESTS >
[2018-10-01T10:30:01,000] [INFO] java.lang.Exception: exception	happened
<pre>at TestPrintStackTrace.f(TestPrintStackTrace.java:3)</pre>	
<pre>at TestPrintStackTrace.g(TestPrintStackTrace.java:7)</pre>	
<pre>at TestPrintStackTrace.main(TestPrintStackTrace.java:16)</pre>	

The following figure shows how the modified regular expression matches the sample log entry that contains two colons (::).

REGULAR EXPRESSION V1 V	1 match, 17 steps (~0ms)
<pre>% \[([^]]+)]\s\[(\w+)]\s([\S\s]+).*</pre>	/ gm 🎮
TEST STRING	SWITCH TO UNIT TESTS >
[2018-10-01T10:30:01,000] [INFO] java.lang.Exception: exception	happened
at TestPrintStackTrace.f(TestPrintStackTrace.java:3) at TestPrintStackTrace.g(TestPrintStackTrace.java7)	

You can perform the preceding steps to debug your regular expression. After you validate the regular expression, you can apply the expression to a Logtail configuration.

4.13.1.5. How do I optimize regular expressions?

A regular expression that accurately matches data with a high match rate can improve the performance of log collection.

When you optimize regular expressions, we recommend that you conform to the following rules:

Use precise characters.

We recommend that you do not use wildcard characters ... in a regular expression to match fields in log entries. Wildcard characters may cause mismatches that reduce the matching performance. For example, if you want to extract a field that consists of only letters, use [A-Za-z].

Use appropriate quantifiers.

We recommend that you do not use +,* . For example, if you want to query data in a more efficient and accurate manner, use \d instead of \d+ or \d(1,3) to match IP addresses.

• Debug regular expressions.

If an error occurs when you use a regular expression, you can use online tools such as Regex101 to debug the regular expression. This way, you can troubleshoot the issue and optimize the regular expression in an efficient manner.

4.13.1.6. How do I use the full regex mode to collect log entries in multiple

formats?

The full regex mode requires that log entries to be collected be in the same format. Therefore, if you want to collect log entries that are in multiple formats, you must use the schema-on-write or schema-on-read solution.

Taking Java logs as an example, the following section lists the types of error log entry and normal log entry.

- Multi-line WARNING log entries
- Simple text INFO log entries
- Key-value DEBUG log entries

[2018-10-01T10:30:31,000] [WARNING] java.lang.Exception: another exception happened

- at TestPrintStackTrace.f(TestPrintStackTrace.java:3)
- at TestPrintStackTrace.g(TestPrintStackTrace.java:7)
- at TestPrintStackTrace.main(TestPrintStackTrace.java:16)
- [2018-10-01T10:30:32,000] [INFO] info something [2018-10-01T10:30:33,000] [DEBUG] key:value key2:value2

To collect log entries of these types, you can use the following solutions:

Schema-on-write: To extract log fields, you must apply multiple Logtail configurations with different regular expressions to a log file.

③ Note However, Logtail cannot apply multiple Logtail configurations directly to the same log file. Therefore, you must set up multiple symbolic links for the directory in which the log file resides. Each Logtail configuration applies to a symbolic link to collect log entries in a specific format.

· Schema-on-read: you can use a common regular expression to collect log entries in different formats.

For example, if you want to collect log entries in multiple formats, you can configure a regular expression that matches the time and log level fields as the first line, and specify the rest of the log entries as the log message. If you want to parse the message, create an index for the message, specify a regular expression to extract log messages, and then extract target fields.

Note We recommend that you use this solution only for scenarios in which tens of millions of log entries are collected, or fewer.

4.13.1.7. How do I specify time formats for logs?

If you configure Logtail to collect logs, you must specify a common time format for the time field of the logs.

- The timestamp of a log is accurate to seconds. Therefore, you can specify the time format only to seconds.
- · You need to specify the time format only for the time in the time field.

The following examples show time formats that are commonly used:

Custom1 2017-12-11 15:05:07 %Y-%m-%d %H:%M:%S Custom2 [2017-12-11 15:05:07.012] [%Y-%m-%d %H:%M:%S] RFC822 02 Jan 06 15:04 MST %d %b %y %H:%M RFC822Z 02 Jan 06 15:04 -0700 %d %b %y %H:%M RFC850 Monday, 02-Jan-06 15:04:05 MST %A, %d-%b-%y %H:%M:%S RFC1123 Mon, 02 Jan 2006 15:04:05 MST %A, %d-%b-%y %H:%M:%S 2006-01-02T15:04:05Z07:00 RFC3339 %Y-%m-%dT%H:%M:%S RFC3339Nano 2006-01-02T15:04:05.999999999207:00 %Y-%m-%dT%H:%M:%S

4.13.1.8. How do I configure non-printable characters in a sample log?

This topic describes how to configure non-printable characters in a sample log.

Background information

If you collect logs in delimiter mode, Log Service allows you to specify a non-printable character as the delimiter or quote. Non-printable characters are characters whose decimal ASCII codes are within the range of 1 to 31 and 127. If you use a non-printable character as a delimiter or quote, you must find the hexadecimal ASCII code of the character and enter the character in the following format: OxHexadecimal ASCII code of the non-printable character and enter the character in the following format: OxHexadecimal ASCII code of the non-printable character and enter the character is the delimiter and 0x02 as the quote, and then enter a non-printable character 0x01 between the digits 5 and 6.

Procedure

- 1. Log on to the Log Service console.
- 2. Right-click the blank space on the browser and select Inspect from the shortcut menu.
- 3. On the page that appears, click the **Console** tab.
- 4. Enter " $\x01$ " in the code editor and press Enter.
- 5. Copy the returned result.
 - A non-printable character is enclosed in double quotation marks (").



6. Paste the returned result between the digits 5 and 6.

In the Logtail Config step, paste the result in the Log Sample field. For more information, see Collect logs in delimiter mode.



4.13.1.9. How do I troubleshoot errors that occur when I collect logs from

containers?

If an error occurs when you use Logtail to collect logs from Docker containers, self-managed Kubernetes, or Container Service for Kubernetes (ACK), you can perform the steps that are described in this topic to troubleshoot the issue.

Troubleshoot an error if Log Service does not receive heartbeats from a Logtail client

To check whether Logtail is installed, perform the following steps:

1. View the heartbeat status of servers in a machine group.

 ${\bf i}$. Log on to the Log Service console

- ii. In the Projects section, click the project that you want to manage
- iii. In the left-side navigation pane, choose Resource > Machine Groups
- iv. In the Machine Groups list, click the name of the machine group that you want to view.
- In the Machine Group Status section, count the number of servers whose heartbeat status is OK.
- 2. Count the number of worker nodes in the related cluster.

cn-hangzhou.i-bplad2b02jtqdlshi2ut Ready

- i. Log on to a master node in the Kubernetes cluster. For more information, see Use kubectl to connect to a Kubernetes cluster in User Guide for Container Service for Kubernetes.
- ii. Run the following command to view the number of worker nodes in the cluster:
 - kubectl get node | grep -v master

The following output is expected:						
NAME	STATUS	ROLES	AGE			
cn=hangzhou_i=bp17enxc2us3624wexb2	Ready	<none></none>	2384			

3. Check whether the number of servers whose heartbeat status is **OK** in the machine group is equal to the number of worker nodes in the cluster. Troubleshoot the error based on the check result.

220d

- $\circ~$ The heartbeat status of all servers in the machine group is $\mbox{\bf Failed}.$
- If you use Logtail to collect standard Docker logs, check whether the values of the \${your_region_name}, \${your_aliyun_user_id}, and \${your_m achine_group_user_defined_id} parameters are valid. For more information, see the Parameters section in Collect standard Docker logs.

VERSION v1.10.4

v1.10.4

- If you use Logtail to collect ACK logs, submit a ticket.
- If you use Logtail to collect logs from self-managed Kubernetes, check whether the values of the {your-project-suffix}, {aliuid}, {access-key-id}, and {access-key-secret} parameters are valid. For more information, see the Parameters section in Collect Kubernetes logs.
- If the value of a parameter is invalid, run the helm del --purge alibaba-log-controller command to delete the installation package and reinstall Logtail.
- The number of servers whose heartbeat status is **OK** is less than the number of worker nodes in the cluster.
- a. Check whether a DaemonSet is manually deployed by using a YAML file.
- Run the kubectl get po -n kube-system -l k8s-app=logtail command to perform the check. If the command returns pod information, a DaemonSet is manually deployed by using a YAML file.
- b. Download the latest version of the Logtail DaemonSet template.
- c. Set the \${your_region_name}, \${your_aliyun_user_id}, and \${your_machine_group_name} parameters based on your business requirements.
- d. Run the kubectl apply -f ./logtail-daemonset.yaml command to update the DaemonSet YAML file.
- If the error persists, submit a ticket to contact Log Service technical support.

Troubleshoot an error if Log Service does not collect logs from containers

If no log is displayed in the **Consumption Preview** panel or on the **Search & Analysis** page of a Logstore, Log Service does not collect logs from the machine group of the Logstore. Check the status of the containers that correspond to the servers in the machine group. If the containers are working as expected, perform the following steps to troubleshoot the error:

- 1. Check the heartbeat status of the servers in the machine group. For more information, see View the heartbeat status of servers in a machine group
- 2. Check whether the parameter settings in the related Logtail configuration are correct.

Check whether the values of the IncludeLabel, ExcludeLabel, IncludeEnv, and ExcludeEnv parameters in the Logtail configuration meet your business requirements.

? Note

The IncludeLabel or ExcludeLabel parameter specifies whether to include the container images to which specified labels are attached. You can run the **docker inspect** command to retrieve a list of container image labels. These labels are not the labels that are defined by using Kubernetes. To check whether the parameter settings are valid in a Logtail configuration, delete the **IncludeLabel**, **ExcludeLabel**, **IncludeEnv**, and **ExcludeEnv** parameters of the Logtail configuration. If Log Service can collect logs from the containers after you delete the parameters, the parameter settings are invalid.

3. Check other items.

Log Service does not collect logs from containers in the following scenarios:

- Log files are not updated.
- The log files of a container are not stored in the default storage or a storage attached to the container.

Other O&M operations

- Log on to a Logtail container
- View the operational logs of Logtail
- Ignore the stdout logs of a Logtail container
- View the status of Log Service components in a Kubernetes cluster
- View the version number, IP address, and startup time of Logtail

Log on to a Logtail container

Use one of the following methods based on your business requirements.

Docker

i. Log on to the host and run the following command to view and record the ID of the Logtail container:

docker ps | grep logtail

ii. Run the following command to log on to the Logtail container:

docker exec -it [\$ID] bash

Note[\$ID] is the ID of the Logtail container

- Kubernetes
- i. Run the following command to view and record the pod where the Logtail container resides:

kubectl get po -n kube-system | grep logtail

ii. Run the following command to log on to the pod:

kubectl exec -it -n kube-system [\$Pod_ID] bash

```
⑦ Note
[$Pod_ID] is the ID of the pod.
```

View the operational logs of Logtail

The operational logs of Logtail are stored in the files named ilogtail.LOG and logtail_plugin.LOG in the /usr/local/ilogtail/ directory of a Logtail container.

- 1. Log on to a Logtail container. For more information, see Log on to a Logtail container.
- 2. Run the following command to go to the /usr/local/ilogtail/ directory:

```
cd /usr/local/ilogtail
```

- 3. Run the following commands in sequence to view the ilogtail.LOG and logtail_plugin. LOG files:
 - cat ilogtail.LOG cat logtail_plugin.LOG

Ignore the stdout logs of a Logtail container

The standard output of the container is irrelevant to this case. Ignore the following standard output:

```
start umount useless mount points, /shm$|/merged$|/mqueu$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c37869500fbe2bdb95d13b1e110172ef57fe840c82155/merged: must be superuser to
umount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749clbf8c16edff44beab6e69718/merged: must be superuser to
umnount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640b1e16c22dbe/merged: must be superuser to
umnount
......
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail is running
```

View the status of Log Service components in a Kubernetes cluster

1. Log on to a master node in the Kubernetes cluster. For more information, see Use kubectl to connect to a Kubernetes cluster in User Guide for Container Service for Kubernetes.

2. Run the following command to view the status of Log Service components in the Kubernetes cluster:

helm status alibaba-log-controller

View the version number, IP address, and startup time of Logtail

- 1. Log on to a Logtail container. For more information, see Log on to a Logtail container.
- 2. Run the following command to view the version number, IP address, and start time of Logtail:

kubectl exec logtail-ds-gb92k -n kube-system cat /usr/local/ilogtail/app_info.json

The following output is expected:

```
{
   "UUID": "",
   "hostname": "logtail-gb92k",
   "instance_id": "OEBB2B0E-0A3B-11E8-B0CE-0A58AC140402_10.10.10.10_1517810940",
   "ip": "203.0.113.10",
   "logtail_version": "0.16.2",
   "os": "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
   "update_time": "2018-02-05 06:09:01"
}
```

4.13.1.10. How do I obtain the labels and environment variables of a container?

Log Service allows you to collect logs from containers. You can specify the containers by label or environment variable. Labels are retrieved by running the docker inspect command and environment variables are specified in the startup configuration of each container.

Obtain container labels

Log on to the host where the container whose labels you want to obtain resides. For example, the host is an Elastic Compute Service (ECS) instance.
 Run the following command to obtain the ID of the container.

The orders variable in the command is the name of a container group. Replace the value of the variable with an actual name.

docker ps | grep orders

2ba4ebdaf503 in the response indicates the ID of the container.

[root@iZbp14up92567	'375kqxjeqZ ~]# docker ps grep orders			
2ba4ebdaf503	43e27feaa78a	"/usr/local/bin/java"	2 months ago	Up 2 month
	k8s_orders_orders-7895d5f946-s6xxj_victor-center_2348cd71-6a91-4b5t	f-af26-73fc03a9c571_0		
0778af9ae173	registry-vpc.cn-hangzhou.aliyuncs.com/acs/pause-amd64:3.0	"/pause"	2 months ago	Up 2 month
	k8s POD orders-7895d5f946-s6xxi victor-center 2348cd71-6a91-4b5f-at	f26-73fc03a9c571 0		

- Run the following command to obtain the labels of the container. The 2ba4ebdaf503 variable in the command is the ID of a container. Replace the value of the variable with an actual ID.

The Labels field in the response indicates the container labels.

"OnBuild": null,
"Labels": {
"annotation.com.aliyun.ack.hashVersion": "1.16.6",
"annotation.io.kubernetes.container.hash": "eabe30b0",
"annotation.io.kubernetes.container.ports": "[{\"containerPort\":80,\"protocol\":\"TCP\"}]",
"annotation.io.kubernetes.container.restartCount": "0",
"annotation.io.kubernetes.container.terminationMessagePath": "/dev/termination-log",
"annotation.io.kubernetes.container.terminationMessagePolicy": "File".
"annotation.io.kubernetes.pod.terminationGracePeriod": "30",
"io.kubernetes.container.logpath": "/var/log/pods/victor-center orders-7895d5f946-s6xxi 2348cd11 - 10 - 10 - 10 - 10 - 10 - 10 - 10 -
"io.kubernetes.container.name": "orders".
"io.kubernetes.docker.tvpe": "container".
"io.kubernetes.pod.name": "orders-7895d5f946-s6xxi".
"io.kubernetes.pod.namespace": "victor-center".
"io.kubernetes.pod.uid": "2348cd71-1011 90 9c571".
"in kubernetes, sandhox, id": "0778af
"msd java build commit": "55998875baar The Transformer 12015b207"
"mad juvi_build_data": "2017-11-2112:52:16:00000"
"msd java build version": "A A 2-SNAPSHOT"
NetworksetLings : {
Bridge:
"Sanadoxiu": ",
Hairpinmode : taise,

Obtain environment variables

- 1. Log on to the host where the container whose labels you want to obtain resides. For example, the host is an ECS instance.
- Run the following command to obtain the ID of the container. The orders variable in the command is the name of a container group. Replace the value of the variable with an actual name.

docker ps | grep orders

2ba4ebdaf503 in the response indicates the ID of the container.

root@iZbp14up925	67375kqxjeqZ ~]# docker ps grep orders			
ba4ebdaf503	43e27feaa78a	"/usr/local/bin/java…"	2 months ago	Up 2 months
	k8s_orders_orders-7895d5f946-s6xxj_victor-center_2348cd71-6a91-4b5	5f-af26-73fc03a9c571_0		
778af9ae173	registry-vpc.cn-hangzhou.aliyuncs.com/acs/pause-amd64:3.0	"/pause"	2 months ago	Up 2 months
	k8s POD orders-7895d5f946-s6xxi victor-center 2348cd71-6a91-4b5f-a	af26-73fc03a9c571 0		

3. Run the following command to obtain the environment variables of the container. The 2ba4ebdaf503 variable in the command is the ID of a container. Replace the value of the variable with an actual ID.

docker exec 2ba4ebdaf503 env

JAVA_OPTS=-Xms64m -Xmx128m -XX:PermSize=32m	-XX:MaxPermSize=64m	-XX:+UseG1GC	-Djava.security.egd=file:/dev/urandom
FRONT_END_SERVICE_HOST=172.2			
PAYMENT_PORT_80_TCP=tcp://172.11.14.5:80			
CATALOGUE_DB_SERVICE_HOST=172 📰 📕.96			
MONGO_SERVICE_PORT_MONGO=27017			
ANTICHEATING_PORT_80_TCP_PROTO=tcp			
CATALOGUE_SERVICE_PORT_CATALOGUE=80			
FRONT_END_PORT_8079_TCP_PROTO=tcp			
FRONT_END_PORT_8079_TCP_ADDR=172			
USER_PORT_80_TCP_PROTO=tcp			
MONGO_PORT_27017_TCP_ADDR=172			
PAYMENT_PORT=tcp://172.@1.100.5:80			
CARTS_PORT_80_TCP=tcp://17			
INTEGRAL_PORT=tcp://172.21.13:80			
CATALOGUE_PORT_80_TCP=tcp://172.21.113:80			
USER_PORT_80_TCP=tcp://172.31 1 101:80			
KUBERNETES_PORT=tcp://172.21 443			
CARTS_PORT=tcp://172 5.30:80			
TEST_PORT_27017_TCP_ADDR=172			
INTEGRAL_PORT_80_TCP_ADDR=171.12.53			
CATALOGUE_SERVICE_HOST=171_21_F_133			
SESSION_DB_PORT=tcp://172 💼 💺 149:6379			
INTEGRAL_SERVICE_PORT_INTEGRAL=80			
RABBITMQ_PORT_5672_TCP_PORT=5672			
RABBITMQ_PORT_5672_TCP_ADDR=172			
ORDERS_SERVICE_PORT=80			
TEST_PORT_27017_TCP_PORT=27017			
CATALOGUE_DB_PORT_3306_TCP=tcp://172	:3306		
INTEGRAL_DB_SERVICE_PORT=3306			
INTEGRAL_SERVICE_PORT=80			
CARTS_SERVICE_HOST=172_11.0,30			
CARTS_SERVICE_PORT=80			
CATALOGUE_PORT_80_TCP_PROTO=tcp			
SESSION DB PORT 6379 TCP PORT=6379			

4.13.2. Log search and analysis

4.13.2.1. FAQ about log query

This topic provides answers to some frequently asked questions (FAQ) about log query in Log Service.

How do I identify the source server from which Logtail collects logs during a query?

docker inspect 2ba4ebdaf503

If a machine group uses IP addresses as its identifiers when logs are collected by using Logtail, the servers in the machine group are distinguished by internal IP addresses. When you query logs, you can use the **hostname** and custom IP address to identify the source server from which logs are collected.

For example, you can use the following query statement to calculate the number of occurrences of each hostname.

(?) Note You must configure an index for the _tag_:_hostname_ field and enable the analysis feature.

* | select "__tag__:__hostname__" , count(1) as count group by "__tag__:__hostname__'

How do I query IP addresses in logs?

You can use the exact match method to query IP addresses in logs. You can search for log data by IP address. For example, you can specify whether to include or exclude an IP address. However, you cannot use the partial match method to query the IP addresses in logs. This is because decimal points contained in an IP address are not default delimiters in Log Service. You can also filter data by using other methods. For example, you can use an SDK to download data and then use a regular expression or the string.indexof() method to search for results.

For example, if you execute the following query statement, the logs that contain the 203.0.113 CIDR block are still returned.

not ip:203.0.113 not status:200 not 360jk not DNSPod-Monitor not status:302 not jiankongbao

not 301 and status:403

How do I query log data by using a keyword that contains a space character?

If you use a keyword that contains a space character to query log data, log data that contains a part of the keyword on the left or right of the space character is returned. You can enclose the keyword that contains a space character in double quotation marks (""). The entire enclosed content is regarded as a keyword to query log data as expected.

For example, you want to query log data that contains the keyword POS version from the following log data:

post():351]: device_id: BTAddr : B6:xF:xx:65:xx:Al IMEI : 35847xx22xx81x9 WifiAddr : 4c:xx:0e:: :xx | user_id: bb07263xxd2axx43xx9exxea26e39e5f POS version:903

If you use POS version as the keyword, log data that contains POS and version is returned. This query result does not meet your requirements. If you use "POS version" as the keyword, log data that contains the keyword POS version is returned.

How do I use two conditions to query log data?

You can specify two conditions in a query statement to query log data.

For example, if you want to query logs in which the value of the status field is not 200 and the value of the request_method field is not GET in a Logstore, you can execute the not status:200 not request_method:GET statement to query logs as expected.

How do I query collected logs in Log Service?

You can use one of the following methods to query logs in Log Service:

- Use the Log Service console.
- Use an SDK.
- Use the Restful API

4.13.2.2. What can I do if I cannot obtain the required results from a log query?

If you cannot find the required log data by using the query feature of Log Service, perform the steps that are described in this topic to troubleshoot the issue.

Log collection failure

If log data fails to be collected by Log Service, you cannot query the log data. Check whether log data is available on the consumption preview page of your Logstore.

If log data is available, log data is collected by Log Service.

If log data is unavailable, check whether the issue occurs due to the following causes:

- The log source does not generate log data.
- If no log data is generated by the log source, no log data can be sent to Log Service. Check your log source.
- Logtail has no heartbeat.

On the **Machine Group Settings** page, check whether the heartbeat status of the related server is OK in the Machine Group Status section. For more information about how to troubleshoot the issue if Log Service does not receive heartbeats from a Logtail client, see What can I do if Log Service does not receive heartbeats from a Logtail client?

· Data is not written to the file that is monitored in real time.

If data is not written to the file that is monitored in real time, you can view error messages in the /usr/local/ilogtail/ilogtail.LOG file. Common error messages:

- parse delimiter log fail: The error message returned because an error occurs when Log Service collects logs in delimiter mode.
- parse regex log fail: The error message returned because an error occurs when Log Service collects logs in full regex mode.

Delimiter setting errors

View the specified delimiters and check whether you can use a keyword to find a log after the log content is split by using the delimiters. In this example, the default delimiters $,;=()[]{2@s<>::'}$ are used. If a log contains abc"defg,hij, the log is split into the following two words: abc"defg and hij. If the log is split, you cannot use the keyword abc to find the log.

Fuzzy match is supported. For more information, see Search syntax.

? Note

- The indexing feature of Log Service is optimized. If you configure both full-text indexes and field indexes, the configurations of the field indexes take precedence. This helps you reduce indexing costs. For example, you configure an index for a log field whose key is message and specify a space character as a delimiter. To use a space character as a delimiter, you must specify the space character in the middle of the delimiters that you specify for an index. You can find a log that contains "message: this is a test message" by using the keyword message: this in the key:value format. However, you cannot find the log by using the keyword this because you have configured a field index for the key and the full-text indexing feature does not take effect for the log field.
- You can create indexes or modify existing indexes. However, new or modified indexes take effect only for new data.
 You can click Index Attributes to check whether the specified delimiters meet your business requirements.

Other reasons

If log data is generated, modify the query time range and perform a query operation again. Log Service allows you to preview log data in real time. The maximum latency of the query feature is 1 minute. We recommend that you query log data at least 1 minute after logs are generated. If the issue persists, submit a ticket.

4.13.2.3. What are the differences between log consumption and log query?

Log Service provides the log consumption and log query features that allow you to read log data from Log Service.

Log consumption

The log consumption feature allows you to read and write full data in the first-in, first-out (FIFO) order. This feature is similar to the features provided by Kafka.

- · Each Logstore has one or more shards. Data is written to a random shard.
- You can read multiple logs at a time from a specified shard based on the order in which the logs were written to the shard.
- You can specify a start position (cursor) to pull logs from shards based on the time when Log Service receives the logs.

Log query

Log Service allows you to query and analyze a large amount of log data based on specific conditions.

- You can specify query conditions to find required log data.
- You can use multiple operators such as AND, NOT, and OR to specify query conditions and perform SQL analysis on query results.
- The log query feature is independent of shards.

Differences

Item	Log query	Log consumption
Search by keyword	Supported.	Not supported.
Data read (a small amount of data)	Fast.	Fast.
Data read (full data)	Slow. Log Service reads 100 logs in 100 milliseconds. We recommend that you do not use this method.	Fast. Log Service reads 1 MB of log data in 10 milliseconds. We recommend that you use this method.
Data read by topic	Yes.	No. Data is identified only by shard.
Data read by shard	No. Data in all shards of a Logstore is queried.	Yes. You need to specify a shard each time to read data.
Fee	Medium.	Low.
Scenario	Monitoring, issue troubleshooting, and analysis.	Full data processing scenarios, such as stream computing and batch processing.

4.13.2.4. How do I resolve common errors that occur when I query log data?

This topic describes the common error messages that are returned when you query log data in the Log Service console and provides related solutions.

Error messages

Error message	Cause	Solution
line 1:44: Column 'XXX' cannot be resolved;please add the column in the index attribute	The XXX key cannot be specified in the query statement because the key does not exist.	Click Index Attributes to configure an index for the field. For more information, see Configure indexes.
ErrorType:QueryParseError.ErrorMessage:syntax error error position is from column:10 to column:11,error near < : >	The query statement contains unnecessary colons (:).	Delete the unnecessary colons (:) from the query statement, and then execute the query statement.
Column 'XXX' not in GROUP BY clause;please add the column in the index attribute	You use a GROUP BY clause and specify a non-GROUP BY field in a SELECT statement. For example, you do not specify the key1 field in the arg (latency) group by key2 GROUP BY clause.	You must specify the same field that you specified in the SELECT statement in the GROUP BY clause. Example: * select key1,avg(latency) group by key1,key2
sql query must follow search query,please read syntax doc	The syntax of the query statement is invalid because a search statement is not specified.	<pre>Invalid query statement: select ip,count(*) group by ip . Valid query statement: * select ip,count(*) group by ip .</pre>
line 1:10: identifiers must not start with a digit; surround the identifier with double quotes	The column name or variable name that is referenced in an SQL statement cannot start with a digit.	Change the column name or variable name to a name that starts with a letter.

line 1:9: extraneous input " expecting	One or more words are misspelled.	Correct the misspelled words.
key (XXX) is not config as key value config,if symbol : is in your log,please wrap : with quotation mark "	The XXX field cannot be referenced in the analytic statement because no field index is configured for the field.	Click Index Attributes to configure an index for the field. For more information, see Configure indexes.
Query exceeded max memory size of 3GB	The size of the memory that is used by the query statement exceeds 3 GB. The issue occurs because a large number of values are returned in the query result after you use a GROUP BY clause to remove duplicates.	Optimize the GROUP BY clause. Reduce the number of keys that is specified in the GROUP BY clause.
ErrorType:ColumnNotExists.ErrorPosition,line:0,colum n:1.ErrorMessage:line 1:123: Column 'XXX' cannot be resolved; it seems XXX is wrapper by ";if XXX is a string ,not a key field, please use 'XXX'	XXX is not an indexed field. In an SQL statement, you must enclose an indexed field in double quotation marks ("") and you must enclose a string in single quotation marks (").	If you want to reference the XXX field, make sure that you index the field and enable the analysis feature for the field. For more information, see Configure indexes. Note If XXX is a string, you must replace the double quotation marks (") with single quotation marks (').
user can only run 15 query concurrently	More than 15 concurrent search statements are executed. A maximum of 15 concurrent search statements can be executed by a user in a project.	Reduce the number of search statements based on your business requirements.
unclosed string quote	The double quotation marks (") in the query statement are incomplete.	Check the query statement, specify double quotation marks (") in pairs, and then execute the query statement.
error after :.error detail:error after :.error detail:line 1:147: mismatched input 'in' expecting { <eof>, 'GROUP', 'ORDER', 'HAVING', 'LIMIT', 'OR', 'AND', 'UNION', 'EXCEPT', 'INTERSECT'}</eof>	The syntax of the query statement is valid.	Modify the SQL statement as prompted and execute the SQL statement.
Duplicate keys (XXX) are notallowed	Indexes are not case-sensitive. For example, if the aBc index exists, an error message is returned when you create the abc index.	Check whether duplicate indexes exist.
only support * or ? in the middle or end of the query	You can specify only asterisks (*) and question marks (?) in the middle or end of a field value to perform a fuzzy match.	You can use the SQL LIKE operator in a query statement based on your business requirements. For example, you cannot use Msg: *xxx to search for logs that contains the Msg field and the field value ends with XXX. You can use the SQL LIKE operator to perform a fuzzy match, as shown in the following query statement: Msg: * SELECT Msg WHERE Msg LIKE '%xxx'
logstore (xxx) is not found	The XXX Logstore does not exist or the analysis feature is not enabled.	Check whether the Logstore exists. If the Logstore exists, you must index at least one field and enable the analysis feature for the Logstore.
condition number 43 is more than 30	A maximum of 30 fields can be referenced by a user in a query statement.	Modify the query statement to reduce the number of the referenced fields, and then execute the query statement.

4.13.2.5. Why data queries are inaccurate?

This topic describes the causes for inaccurate data queries. It also includes solutions to these issues.

When you search and analyze log data, the message **The results are inaccurate** may prompt in the console. This indicates that the returned result is inaccurate because some log data in a Logstore was not queried.

Possible causes include:

The time range for queries is excessive.

Cause

The time range for a query is excessively wide, for example, three months or a year. In this case, Log Service cannot scan all log data generated within this time period for one query.

Solution

Narrow down the query time range and perform multiple queries.

Query statements are complex.

Cause

The query statement is exceedingly complex or contains multiple frequently used words. In this case, Log Service cannot scan all related log data or read the query results at one time.

Solution

Narrow down the query scope and perform multiple queries.

The SQL computing needs to read an excessively large amount of data.

Cause

The SQL computing needs to read an excessively large amount of data. In this case, query results are likely to become inaccurate. A maximum of 1 GB of data can be read from each shard. For example, if the SQL computing needs to read strings from multiple columns, which exceed the threshold data volume, inaccurate query results will be returned.

Solution

Narrow down the query scope and perform multiple queries.

4.13.2.6. How do I configure indexes for historical log data?

You cannot directly configure indexes for historical log data in Log Service. To configure indexes for historical log data, you can use DataWorks or the command line interface (CLI) to move data into another Logstore.

Indexes take effect on log data that is collected after the indexes are configured. You cannot use indexes to search and analyze historical log data. To configure indexes for historical log data, you can use either of the following two methods:
Configure indexes in a new Logstore and then use DataWorks to move data into the Logstore.

- After you configure indexes in a new Logstore, you can use DataWorks to move historical log data from the Logstore where it is stored to the new Logstore. Then you can use the configured indexes to search and analyze the data.
- After you configure indexes in a new Logstore, you can use the CLI to export historical log data from the Logstore where it is stored to the new Logstore.

(2) Note The preceding two methods copy historical log data and then export the data into a Logstore. They do not change or delete the data.

4.13.3. Alarm

4.13.3.1. FAQ about alerts

This topic provides answers to some frequently asked questions (FAQ) about alerts in the Log Service console.

How do I add raw logs to an alert notification?

For example, if more than five error logs are detected in the previous 5 minutes, an alert is triggered and an alert notification is sent. To add the raw logs to the alert notification, perform the following steps:

1. Log on to the Log Service console

- 2. In the Projects section, click the project that you want to manage.
- 3. In the left-side navigation pane, click the $rac{1}{100}$ icon.
- 4. In the Dashboard list, click the dashboard for which you want to configure alert rules.
- 5. In the upper-right corner of the chart, choose || > **Create Alert**.

6. In the Create Alert panel, set the parameters. The following figure shows the parameter settings.

Modify Alert					\times	Modify Alert				\times
Alert	Configu	uration	Not	ifications		Alert Conf	iguration	Notificat	ions	
* Alert Name	alarn	n_test		~ =	á	Notifications		Email ×		\sim
 Associated Chart 	•	Chart Name	test-pie-chart	~ @		✓ Email)	×
		Query	level: ERROR		í.	* Recipients	abc@test.com		12/256	
		Search Period	① 1Hour(Time Frame) ▼			Subject	Use commas (,) to separate mu	ultiple recipients.	17/128	
	1	Chart Name	chart-01	~ @		* Content	\${results[0].rawresults}		17/120	
		Query	level: ERROR select COUNT(*)	as count	á					
		Search Period	① 1Hour(Time Frame) ▼							
	2	- Add				5	Supported template variables:	\${Project}, \${Condition}, \${	AlertName},	
* Search Interval	15	+	Minutes 🗸			ŝ	\${AlertID}, \${Dashboard}, \${Fire	eTime}, \${Results} View all	variables	
 Trigger Condition (2) 	\$1.c	ount>5								
	Suppo (%) op I~.Doc	rt the addition (+ erations and co umentation	 -), subtraction (-), multiplication (mparison operations including >, 	*), division (/), and modulo >=, <, <=, ==, !=, =~, and	1					
			I	Next Cance	l.			Previous Sub	mit Can	ncel

The following example shows how to set the parameters:

Query

- Association Chart 0: level:ERROR
- Association Chart 1: level: ERROR | select COUNT(*) as count
- Trigger Condition: **\$1.count > 5**
- o Content: \${results[0].rawresults}

7. Click Submit.

What can I do if the DingTalk chatbot fails to send a notification?

For example, after you set the notification method to WebHook-DingTalk Bot, the following error message is returned when the DingTalk chatbot sends a notification:

{"errcode":310000,"errmsg":"sign not match"}
{"errcode":310000,"errmsg":"keywords not in content"}

This error message is returned because the security settings of the latest chatbot are invalid. You can reconfigure the security settings. For more information, see WebHook-DingTalk Bot. If the issue persists or other error messages are returned, submit a ticket.

4.13.4. What do I do if the Forbidden.SLS::ListProject error occurs when I

log on to the Apsara Uni-manager Management Console?

Symptoms

If you are redirected to the Log Service console when you use a RAM user to log on to the Apsara Uni-manager Management Console, the Forbidden.SLS::ListProject error may occur.

Causes

The Log Service-related role that the RAM user assumes is not granted the permissions to access Log Service. Therefore, you cannot use the RAM user to access Log Service.

Solutions

- 1. Log on to the Apsara Uni-manager Management Console.
- 2. In the top navigation bar, click **Enterprise**.
- 3. In the left-side navigation pane, choose Permissions > Role Permissions. In the search box, enter SLS to view the role.
- 4. Click **Modify** in the Actions column.
- 5. On the details page, click the **Application Permissions** tab.
- 6. Select View resources in the Log Service section and click Update.
- 7. Use the RAM user to log on to the Apsara Uni-manager Management Console again.

5.Tablestore 5.1. Terms

This topic describes several basic terms used in Tablestore, including data model, max versions, time to live (TTL), max version offset, primary key and attribute, read/write throughput, region, instance, endpoint, and Serial ATA (SATA).

data model

A data model that consists of tables, rows, primary keys, and attribute columns in Tablestore. The following figure shows an example of a data model.



max versions

A data table attribute that indicates the maximum number of data versions that can be stored in each attribute column of a data table. If the number of versions in an attribute column exceeds the max versions value, the earliest version is asynchronously deleted.

TTL

A data table attribute that indicates the validity period of data in seconds. To save space and reduce costs for data storage, Tablestore deletes any data that exceeds its TTL.

max version offset

A data table attribute that describes the maximum allowable difference between the version to be written and the current time in seconds.

To prevent the writing of unexpected data, a server checks the versions of attribute columns when the server processes writing requests. If the specified version is earlier than the current writing time minus the max version offset value or later than or equal to the current writing time plus the max version offset value, data fails to be written to the row.

The valid version range of an attribute column: [max{Data written time - Max version offset, Data written time - TTL value}, Data written time + Max version offset). Data written time is the number of seconds that have elapsed since 00:00:00, 1 January 1970. Versions of the attribute columns are written in milliseconds. A version of an attribute column must fall within the valid version range after the version number is converted to seconds (divide by 1,000).

primary key and attribute

A primary key is the unique identifier of each row in a table. A primary key consists of one to four primary key columns. When you create a table, you must define a primary key. You must specify the name, data type, and sequence of each primary key column. The data type of primary key columns can be only STRING, INTEGER, or BINARY. The size of a STRING or BINARY primary key column cannot exceed 1 KB.

An attribute is the attribute data stored in a row. You can create an unlimited number of attribute columns for each row.

read/write throughput

A Tablestore attribute that is measured by read/write capacity units (CUs).

region

An Apsara Stack physical data center. Tablestore is deployed across multiple Apsara Stack regions. Select a region that suits your business requirements.

instance

A logical entity that is used to manage tables in Tablestore. Instances correspond to databases in traditional relational databases. An instance is the basic unit of the Tablestore resource management system. Tablestore allows you to control access and meter resources by instance.

endpoint

The connection URL for each instance. You must specify an endpoint before you perform any operations on Tablestore tables and data.

SATA

A disk that is based on serial connections and provides stronger error-correcting capabilities. Serial ATA aims to improve the reliability of data during transmission.

5.2. Limits

This topic describes the limits of which you need to take note when you use Tablestore.

User Guide Tablestore

The following table describes the limits on the usage of Tablestore. Some limits indicate the maximum values that you can specify rather than the recommended values. You can tailor the table schemas and row sizes to improve performance.

Item	Limit	Description
Number of instances created within an Alibaba Cloud account	1,024	If you need to increase the maximum number of instances, contact the administrator.
Number of tables in an instance	1,024	If you need to increase the maximum number of tables, contact the administrator.
Length of an instance name	3 to 16 bytes	The name of an instance can contain letters, digits, and hyphens (-). The name must start with a letter and cannot end with a hyphen (-).
Length of a table name	1 to 255 bytes	The name of a table can contain letters, digits, and underscores (_). The name must start with a letter or an underscore (_).
Length of a column name	1 to 255 bytes	The column name can contain letters, digits, and underscores (_). The name must start with a letter or an underscore (_).
Number of columns in a primary key	1 to 4	A primary key can contain one to four primary key columns.
Size of the value in a primary key column of the STRING type	1 KB	The size of the value in a primary key column of the STRING type can be up to 1 KB.
Size of the value in an attribute column of the STRING type	2 MB	The size of the value in an attribute column of the STRING type can be up to 2 MB.
Size of the value in a primary key column of the BINARY type	1 KB	The size of the value in a primary key column of the BINARY type can be up to 1 KB.
Size of the value in an attribute column of the BINARY type	2 MB	The size of the value in an attribute column of the BINARY type can be up to 2 MB.
Number of attribute columns in a single row	Unlimited	A single row can contain an unlimited number of attribute columns.
Number of attribute columns written by one request	1,024	During a PutRow, UpdateRow, or BatchWriteRow operation, the number of attribute columns written in a row can be up to 1,024.
Size of a single row	Unlimited	Tablestore does not impose limits on the total size of the column names and column values for a row.
Number of columns specified by the columns_to_get parameter in a read request	0 to 128	The maximum number of columns returned from a row of data in a read request cannot exceed 128.
Count of UpdateTable operations on a single table	Upper limit: unlimited Lower limit: unlimited	The limit on the count of UpdateTable operations on a single table follows the limit on the frequency of calling the UpdateTable operation.
Frequency of calling the UpdateTable operation on a single table	Once every 2 minutes	The reserved read/write throughput for a table can be adjusted once every two minutes at most.
Number of rows read by one BatchGetRow operation	100	None
Number of rows written by one BatchWriteRow operation	200	None
Size of data written by one BatchWriteRow operation	4 MB	None
Data returned by one GetRange operation	5,000 rows or 4 MB	The amount of data returned by a GetRange operation can be up to 5,000 rows or 4 MB. When either of the limits is exceeded, data that exceeds the limits is truncated at the row level. The primary key information of the next row is returned.
Data size of an HTTP request body	5 MB	None

5.3. Quick start

5.3.1. Log on to the Tablestore console

This topic describes how to log on to the Tablestore console.

Prerequisites

Cloud Defined Storage

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel.
- A web browser is available. We recommend that you use Google Chrome.

Procedure

1. Enter the URL of the Apsara Uni-manager Management Console in the address bar and press the Enter key.

? Note

• You can click the current language in the upper-right corner to switch to another language.

2. Enter your username and password.

Obtain the username and password from an operations administrator.

- ⑦ Note
 - First logon

The first time that you log on to the Apsara Uni-manager Management Console, you need to change the password of your account. The password must be 10 to 32 characters in length. It must contain all of the following character types: uppercase letters, lowercase letters, digits, and special characters. Supported special characters include: ! @ # \$%

Forget password

If you have forgotten your password, click Forgot Password. On the page that appears, enter the username of your account, the email address that was used to create the account, and the CAPTCHA code. Then, the system sends a link for resetting the password to the specified email address.

3. Click Log On.

4. If multi-factor authentication (MFA) is enabled for your account, perform the corresponding operations in the following scenarios:

- $\,\circ\,$ You log on to the Apsara Uni-manager Management Console for the first time after MFA is enabled by the administrator.
 - a. On the Bind Virtual MFA Device page, bind an MFA device.
 - b. Enter the username and password again as in Step 2 and click Log On.
 - c. Enter a six-digit MFA verification code and click Authenticate
- You have enabled MFA and bound an MFA device.
 - Enter a six-digit MFA verification code and click Authenticate.

? Note

For information about how to bind and enable MFA, see Manage MFA in Apsara Uni-manager Management Console User Guide .

5. In the top navigation bar, choose Products > Storage > Tablestore.

5.3.2. Create a Tablestore instance

A Tablestore instance is a logical entity in Tablestore that allows you to use and manage Tablestore services. An instance is the basic unit of resource management in Tablestore. Tablestore controls access requests from applications and collects statistics about the resources that are used by applications deployed in an instance. This topic describes how to create a Tablestore instance.

Procedure

- 1. Log on to the Tablestore console
- 2. On the Overview tab, click Create Instance

? Note

You can create different instances to manage the associated tables in different business scenarios. You can also create different instances for development, testing, and production environments in the same business scenario. By default, Tablestore allows you to create up to 1,024 instances and up to 1,024 tables in each instance within an Alibaba Cloud account.

3. On the Create Instance page, configure the parameters. The following table describes the parameters.

Parameter	Description
Organization	The organization to which the instance belongs.
Resource Set	The resource set to which the instance belongs.
Region	The region in which the instance resides.
Instance Name	The name of the instance. The name of an instance must be 3 to 16 characters in length and can contain only letters, digits, and hyphens (-). The name must start with a letter and cannot start with case-insensitive string ali or ots.
Description	The description of the instance.

Target Cluster	The cluster to which the instance belongs. Tablestore allows you to create an instance in the specified cluster. () Important After you select a cluster, you cannot specify the instance type. This is because the instance type is determined by the storage type of the cluster.
	The type of the instance. Tablestore provides high-performance instances and capacity instances. The instance type varies
Instance Specification	based on the storage type of the cluster to which the instance belongs.

- 4. Click Submit.
- 5. In the **Operation succeeded** message, click **OK**.
 - On the **Overview** page, you can view the instance that you create.
 - Click the name of the instance or click Manage Instance in the Actions column of the instance. On the Instance Management page, you can click different tabs to perform different operations.
 - •
 - To release the instance, click **Release** in the **Actions** column.

() Important

- Before you release an instance, make sure that all data tables in the instance are deleted, and that virtual private clouds (VPCs) are unbound from the instance.
- To prevent conflicts, make sure that the name of an instance that you want to create is different from the name of the instance that is being released.

What to do next

After an instance is created, you can perform operations on the instance. The following table describes the operations that are supported.

Operation	Description
Manage an instance	 After an instance is created, you can view the detailed information about the instance, view the monitoring data, and configure VPCs for the instance. On the Overview page, click the name of the instance that you want to manage. On theInstance Management page, click different tabs to perform different operations. On the Instance Details tab, you can view the endpoints that are used to access the instance and the basic information about the instance. You can also create data tables for the instance and manage the data tables that you create. On the Instance Monitoring tab, you can configure the Time Range, Metric Group, and Operation Type parameters to view the data of monitoring metrics. On the Network Management tab, you can bind or unbind VPCs and view the list of VPCs that are bound to the instance.
Release an instance	 Important Before you release an instance, make sure that all data tables in the instance are deleted, and that VPCs are unbound from the instance. To prevent conflicts, make sure that the name of an instance that you want to create is different from the name of the instance that is being released. You can release an instance if the instance is no longer used. On the Overview page, find the instance that you want to release and clickRelease in the Actions column. In the Release message, confirm the information about the instance. Then, clickOK.

5.3.3. Create a data table

This topic describes how to create a data table in the Tablestore console.

Procedure

- 1. Log on to the Tablestore console
- 2. On the **Overview** page, click the name of the instance for which you want to create a data table or click **Manage Instance** in the **Actions** column of the instance.
- 3. In the Tables section of the Instance Details tab, click Create Table.

?	Note								
You	can create up	to	1,024	data	tables	for	each	instan	ce.

4. In the Create Table dialog box, configure the parameters. The following table describes the parameters.

Parameter

Description

Table Name	The name of the data table. This name is used to identify a data table in an instance. The name must be 1 to 255 bytes in length and can contain letters, digits, and underscores (_). The name must start with a letter or an underscore (_).
Primary Key	One or more primary key columns in the data table that uniquely identify each row in the table. Enter a name for the primary key column and select a data type. To add a primary key column, click Add Primary Key Column . You can specify one to four primary key columns. The first primary key column is the partition key. After you create a data table, you cannot modify the configurations and order of the primary key columns.
	 In Tablestore, only one primary key column can be specified as an auto-increment primary key column for each data table. You cannot specify the partition key as an auto-increment primary key column. After you specify a primary key column as an auto-increment primary key column, Tablestore automatically generates a value for the auto-increment primary key column when you write a row of data. The values of the auto-increment primary key column and unique within the rows that share the same partition key.
	 Naming conventions for primary key columns: The name must be 1 to 255 bytes in length and can contain letters, digits, and underscores (). The name must start with a letter or an underscore (). The STRING, INTEGER, BINARY, and Auto Increment data types are supported by primary key columns.
	You can select Auto Increment as the data type only for a primary key column that is not the partition key.
Allow Updates	Specifies whether to allow data writes by using the UpdateRow operation. Default value: Yes , which indicates that the UpdateRow operation is allowed to write data. If you want to use the time to live (TTL) feature of secondary indexes or search indexes, you must set this parameter t No , which indicates that the UpdateRow operation is not allowed to write data.

5. Optional. Configure the advanced parameters.

If you need to configure the advanced parameters such as Time to Live and Max Versions, perform the following operations:

i. Turn on Advanced Settings.

ii. Configure the advanced parameters. The following table describes the parameters.

Parameter	Description
Time to Live	The duration during which the data in the data table can be retained. If the retention period exceeds the TTL value, Tablestore automatically deletes expired data. The minimum value is 86400 seconds, which is one day. A value of -1 specifies that the data never expires.
Max Versions	The maximum number of versions that can be retained for data in attribute columns of the data table. If the number of versions of data in attribute columns exceeds the value of this parameter, Tablestore deletes the data of earlier versions to keep the number of versions equal to the value of this parameter. You must set this parameter to a value other than 0. Important If you want to create an index table for the data table, you must set the Max Versions parameter to 1.
Max Version Offset	The maximum difference between the current system time and the specified data version. The difference between the version number and the time at which the data is written must be less than or equal to the value of the Max Version Offset parameter. Otherwise, an error occurs when the data is written. The valid version range of data in an attribute column is calculated by using the following formula: Valid version range = [max{Data written time - Max version offset, Data written time - TTL value}, Data written time + Max version offset].

6. Optional. Create a secondary index.

The secondary index feature allows you to create one or more index tables for a data table. Then, you can query data based on the primary key columns of the index tables instead of the data table. This improves query efficiency. If you need to create a secondary index, perform the following operations:

i. Turn on Global Secondary Index.

ii. Click Add in the Pre-defined Column section. Specify a name for the predefined column and select a data type from the drop-down list.

After you add a predefined column to the data table, the predefined column can be used as an attribute column or a primary key column of the index table.

? Note

Tablestore uses a schema-free model. You can write any columns to a row without the need to specify the schema. When you create a data table, you can also predefine columns and specify the data types of the predefined columns.

• You can add up to 20 predefined columns for a data table. To delete a predefined column that you add, click the 💼 icon on the right of the predefined column.

- The name of a predefined column must be 1 to 255 bytes in length and can contain letters, digits, and underscores (_). The name must start with a letter or an underscore (_).
- The following data types are supported for predefined columns: STRING, INTEGER, BINARY, FLOAT, and BOOLEAN.

iii. Click Add Global Secondary Index. Configure the Index Name, Primary Key, Pre-defined Column, and Index Type parameters.

Important

The index name must be different from the data table name. Both the index name and the data table name must be unique in the instance.

- The name of a secondary index must be 1 to 255 bytes in length and can contain letters, digits, and underscores (_). The name must start with a letter or an underscore (_).
- You can specify one or more primary key columns or predefined columns of the data table for the Primary Key parameter.
- You can specify only one or more predefined columns of the data table for the **Pre-defined Column** parameter. However, you cannot specify the same predefined column as a primary key column and a predefined column of the secondary index.
- You can set the Index Type parameter to Global or Local.

If you use the global secondary index feature, Tablestore automatically synchronizes data from the indexed columns and primary key columns of a data table to an index table in asynchronous mode. The synchronization latency is within a few milliseconds. If you use the local secondary index feature, Tablestore automatically synchronizes data from the indexed columns and primary key columns of a data table to an index table in synchronous mode. After data is written to the data table, you can immediately query the data from the index table.

() Important

The first primary key column of a global secondary index can be a primary key column or predefined column of the data table. The first primary key column of a local secondary index must be the first primary key column of the data table.

7. Click **OK**.

After a data table is created, you can view the data table in the **Tables** section. If the data table that you created is not displayed, click the c icon to refresh the table list

What to do next

After a data table is created, you can perform operations on the data table. The following table describes the operations that are supported.

Operation	Description
Manage a data table	 After a data table is created, you can view the detailed information about the data table, manage the data in the table, manage the secondary indexes and search indexes created for the data table, use real-time data consumption tunnels, and view the monitoring metrics of the data table or indexes. Click the name of the data table that you want to manage. On the Manage Table page, click different tabs to perform different operations. On the Basic Information tab, you can view the basic information, advanced features, and primary key columns of the data table or disable the Stream feature. On the Query Data tab, you can insert rows of data, update data, query data by using the primary key or secondary indexes of the data table, retrieve data by using search indexes, view data details, and delete multiple data records at a time. On the Indexes tab, you can create secondary indexes or search indexes, view index details, use indexes to query data, and delete indexes. On the Tunnels tab, you can enable the Stream feature, create and delete tunnels, and view the channels of a tunnel. On the Monitoring Indicators tab, you can configure the Time Range, Metric Group, and Operation Type parameters to view the data of monitoring metrics of the table or indexes.
	The monitoring metrics include service monitoring overview, average access latency, queries per second (QPS), number of rows, and traffic statistics.
Delete a data table	 Important If you delete a data table, the table and the data in the table are permanently deleted from Tablestore and cannot be restored. Proceed with caution. Important 圖除数据表时,系统会永久删除数据表及表中数据,且删除后的数据表及表中数据无法恢复,请谨慎操作。 圖除数据表前请确保已删除创建的索引,否则数据表将删除失败。 You can delete a data table if the data table is no longer used.
	 Find the data table that you want to delete. Click the icon in the Actions column, and then click Delete. In the Delete Table message, confirm the information about the data table. Then, clickOK.

5.3.4. Read and write data in the console

After a data table is created, you can read data from and write data to the data table in the Tablestore console.

Write data

- 1. Log on to the Tablestore console
- 2. On the Overview page, click the name of the instance that you want to manage or click Manage Instance in the Actions column of the instance.
- 3. In the **Tables** section on the **Instance Details** tab, click the name of the data table that you want to manage. You can also click**Query** in the **Actions** column. On the Manage Table page, click the **Query Data** tab.
- 4. On the Query Data tab, click Insert.
- 5. In the **Insert** dialog box, configure the Primary Key Value parameter. Click **Add Column**. Configure the **Name**, **Type**, **Value**, and **Version** parameters.

? Note

If you have created predefined columns for the data table, the predefined columns are displayed in the Attribute Columns section. In this case, you need to only configure the **Value** and **Version** parameters.

By default, **System Time** is selected. This specifies that the current system time is used as the version number of the data. You can also clear **System Time** and specify a version number of the data.

6. Click **OK**.

Rows that are written to the table are displayed on the Query Data tab.

Update data

You can update data in the attribute columns of a row.

- 1. Log on to the Tablestore console
- 2. On the Overview page, click the name of the instance that you want to manage or click Manage Instance in the Actions column of the instance.
- In the Tables section on the Instance Details tab, click the name of the data table that you want to manage. On the Manage Table page, click the Query Data tab. You can also click Query in the Actions column of the data table.
- 4. On the Query Data tab, select the row of data that you want to update. Click Update.
- 5. In the **Update** dialog box, add or delete attribute columns, or update or delete the data in attribute columns.
 - $\circ~$ To add an attribute column, click ${\bf Add~Column}.$
 - To delete an attribute column, click the n icon.
 - In the first Actions column, if you select Update, you can modify the data in attribute columns. If you select Delete, you can delete the data of the selected version. If you select Delete All, you can delete all versions of the data.
- 6. Click **OK**.

Read data

You can query a single row of data or data within the specified range in the Tablestore console.

To query a single row of data, perform the following steps:

- 1. Log on to the Tablestore console
- 2. On the Overview page, click the name of the instance that you want to manage or click Manage Instance in the Actions column of the instance.
- In the Tables section on the Instance Details tab, click the name of the data table that you want to manage. On the Manage Table page, click the Query Data tab. You can also click Query in the Actions column of the data table.
- 4. On the Query Data tab, click Search.
- 5. Specify the query conditions.
- i. In the Search dialog box, set the Modes parameter to Get Row.
- ii. Select a data table or a secondary index.
- iii. By default, the system returns all columns. To return specific attribute columns, turn off **All Columns** and specify the names of the attribute columns that you want to return. Separate multiple attribute columns with commas (,).
- iv. Configure the **Primary Key Value** parameter of the row that you want to query. The integrity and accuracy of the primary key values affect the query results.
- v. Configure the Max Versions parameter to specify the maximum number of versions of the data to return.

? Note

If you use the Tablestore console to read data, the number of returned data versions can be up to 20. If you use a Tablestore SDK to read data, the number of returned data versions is not limited.

6. Click **OK**.

- To query data within the specified range, perform the following steps:
- 1. Log on to the Tablestore console
- 2. On the Overview page, click the name of the instance that you want to manage or click Manage Instance in the Actions column of the instance.
- In the Tables section on the Instance Details tab, click the name of the data table that you want to manage. On the Manage Table page, click the Query Data tab. You can also click Query in the Actions column of the data table.
- 4. On the Query Data tab, click Search.
- 5. Specify the query conditions.
 - i. In the Search dialog box, set the Modes parameter to Range Search.
- ii. Select a data table or a secondary index.
- iii. By default, the system returns all columns. To return specific attribute columns, turn off **All Columns** and specify the names of the attribute columns that you want to return. Separate multiple attribute columns with commas (,).

iv. Configure the Start Primary Key Column and End Primary Key Column parameters.

You can set the Start Primary Key Column parameter to Min Value or Custom and set the End Primary Key Column parameter to Max Value or Custom. If you select Custom, specify a custom value.

? Note

- The GetRange operation follows the leftmost matching principle. Tablestore compares values in sequence from the first primary key column to the last primary key column to read data whose primary key values are in the specified range. For example, the primary key of a data table consists of the following primary key columns: PK1, PK2, and PK3. When data is read, Tablestore first determines whether the PK1 value of a row is in the range that is specified for the first primary key column. If the PK1 value of a row is in the range, Tablestore stops determining whether the values of other primary key columns of the row are in the ranges that are specified for each primary key column and returns the row. If the PK1 value of a row is not in the range, Tablestore continues to determine whether the values of other primary key columns of the row are in the same manner as PK1.
- The range is a left-open and right-closed interval.
- v. Configure the Max Versions parameter to specify the maximum number of versions of the data to return.

? Note

If you use the Tablestore console to read data, the number of returned data versions can be up to 20. If you use a Tablestore SDK to read data, the number of returned data versions is not limited.

vi. Specify the sorting order of the query results. You can select Forward Search or Backward Search.

6. Click OK

Delete data

You can delete the data that you no longer need.

- 1. Log on to the Tablestore console
- 2. On the Overview page, click the name of the instance that you want to manage or click Manage Instance in the Actions column of the instance.
- In the Tables section on the Instance Details tab, click the name of the data table that you want to manage. On the Manage Table page, click the Query Data tab. You can also click Query in the Actions column of the data table.
- 4. On the Query Data tab, select the rows of data that you want to delete. Then, click Delete.
- 5. In the **Delete** message, click **OK**.

5.3.5. Bind a VPC to a Tablestore instance

After you bind a virtual private cloud (VPC) to a Tablestore instance, you can access the Tablestore instance from Elastic Compute Service (ECS) instances in the VPC if the Tablestore instance and the ECS instances reside in the same region.

Prerequisites

- A VPC is created and deployed in the same region as the Tablestore instance.
- One or more ECS instances are created in the VPC.

Procedure

- 1. Log on to the Tablestore console
- 2. On the **Overview** page, click the name of the instance that you want to manage or click **Manage Instance** in the **Actions** column of the instance.
- 3. On the Instance Management page, click the Network Management tab.
- 4. On the Network Management tab, click Bind VPC.
- 5. In the **Bind VPC** dialog box, configure the VPC ID, VSwitch, and **VPC Name** parameters.
- The name of a VPC can contain only letters and digits, and must start with a letter. The name must be 3 to 16 bytes in length.
- 6. Click **OK**.

After the VPC is bound to the instance, you can view the information about the VPC in the **VPCs** section on the **Network Management** tab. You can use the VPC endpoint to access the Tablestore instance from ECS instances in the VPC.

What to do next

After you bind a VPC to a Tablestore instance, you can perform the following operations based on your business requirements:

- Click Details in the Actions column to view the detailed information about the VPC. The information includes the name of the Tablestore instance to
 which the VPC is bound, the name of the VPC, and the VPC endpoint.
- Click Unbind in the Actions column to unbind the VPC from the Tablestore instance. After the VPC is unbound, ECS instances in the VPC can no
 longer access the Tablestore instance over the VPC. To access the Tablestore instance from the ECS instances, you must bind the VPC to the
 Tablestore instance again.

() Important

After you unbind a VPC from a Tablestore instance, you cannot use the VPC endpoint to access the Tablestore instance. Proceed with caution.

5.3.6. Use Tunnel Service

After the Stream feature is enabled for a data table, you can create tunnels for the data table to consume historical and incremental data in the data table.

Background information

Tunnel Service is an integrated service for the consumption of full and incremental data based on the Tablestore API. You can create full, incremental, and differential tunnels to consume data that is stored in a distributed manner in real time.

Enable the Stream feature

Important

The Stream feature is used to implement underlying streaming operations. The Tunnel Service and search index features depend on the Stream feature. In most cases, you do not need to modify or disable the Stream feature. If the Stream feature is disabled for a data table, all existing Stream records of the data table are permanently deleted, and the Tunnel Service and search index features that depend on the Stream feature are also disabled.

After the Stream feature is enabled for a data table, the system periodically deletes the expired Stream operation logs that have been stored longer than the specified period

1. Log on to the Tablestore console.

- 2. On the Overview page, click the name of the instance that you want to manage or click Manage Instance in the Actions column of the instance.
- 3. In the Tables section on the Instance Details tab, click the name of the data table that you want to manage. On the Manage Table page, click the

Tunnels tab. You can also click the 🔹 icon in the **Actions** column of the data table. Then, click **Tunnels**.

4. In the Stream Information section on the Tunnels tab. click Enable.

- 5. In the Enable Stream dialog box, configure the Log Expiration Time parameter.
 - ⑦ Note
 - The Log Expiration Time parameter specifies the duration after which Stream operation logs expire.
 - · Unit: hours. The value of the Log Expiration Time parameter must be a non-zero integer and cannot be modified after it is specified. The duration can be up to 168 hours

6 Click Enabled

Create a tunnel

After you create a tunnel for a data table, you can use the tunnel to consume historical and incremental data in the data table.

- 1. Log on to the Tablestore console
- 2. On the Overview page, click the name of the instance that you want to manage or click Manage Instance in the Actions column of the instance.

3. In the Tables section on the Instance Details tab, click the name of the data table that you want to manage. On the Manage Table page, click the Tunnels tab. You can also click the 👔 icon in the **Actions** column of the data table. Then, click **Tunnels**.

- 4. On the Tunnels tab. click Create Tunnel.
- 5. In the Create Tunnel dialog box, configure the Tunnel Name and Type parameters. Tunnel Service provides the following types of tunnels to consume data that is stored in a distributed manner in real time: Incremental, Full, and Differential.
- 6. Click **OK**.

After the tunnel is created, you can view the information about the tunnel on the **Tunnels** tab.

Preview the data format of a tunnel

- You can simulate data consumption by writing or deleting data and preview the data format of a tunnel.
- 1. Write or delete data. For more information, see Read and write data in the console.
- 2. Preview the data format of a tunnel.
- i. Log on to the Tablestore console
- ii. On the Overview page, click the name of the instance that you want to manage or click Manage Instance in the Actions column of the instance.
- iii. In the Tables section on the Instance Details tab, click the name of the data table that you want to manage. On the Manage Table page, click the

Tunnels tab. You can also click the 👔 icon in the Actions column of the data table. Then, click Tunnels.

- iv. On the Tunnels tab, find the tunnel that you want to manage. Click Show Channels in the Actions column
- v. Find the channel in which you want to simulate data consumption. Click View Simulated Export Records in the Actions column. In the dialog box that appears, click Start.



Enable data consumption for a tunnel

You can use a Tablestore SDK in any programming language for Tunnel Service to enable data consumption for a tunnel by using the ID of the tunnel.

User Guide Tablestore

1. Obtain the ID of the tunnel.

i. Log on to the Tablestore console.

- ii. On the Overview page, click the name of the instance that you want to manage or click Manage Instance in the Actions column of the instance.
- iii. In the Tables section on the Instance Details tab, click the name of the data table that you want to manage. On the Manage Table page, click the

Tunnels tab. You can also click the 👔 icon in the **Actions** column of the data table. Then, click **Tunnels**.

iv. On the **Tunnels** tab, copy the ID of the tunnel.

2. Use a Tablestore SDK in any programming language for Tunnel Service to enable data consumption for the tunnel.

```
// Specify the callback for data consumption to call the IChannelProcessor operation, which indicates the process and shutdown methods.
private static class SimpleProcessor implements IChannelProcessor {
   @Override
    public void process(ProcessRecordsInput input) {
        System.out.println("Default record processor, would print records count");
        System.out.println(
            String.format("Process %d records, NextToken: %s", input.getRecords().size(), input.getNextToken()));
        try {
           // Mock Record Process.
            Thread.sleep(1000);
        } catch (InterruptedException e) {
           e.printStackTrace();
        }
   @Override
   public void shutdown() {
       System.out.println("Mock shutdown");
}
// Specify advanced parameters in the TunnelWorkerConfig class.
TunnelWorkerConfig config = new TunnelWorkerConfig(new SimpleProcessor());
\ensuremath{{\prime}}\xspace // Specify the TunnelWorker parameter and start automatic data processing.
TunnelWorker worker = new TunnelWorker($tunnelId, tunnelClient, config);
try {
    worker.connectAndWorking();
} catch (Exception e)
    e.printStackTrace();
    worker.shutdown();
    tunnelClient.shutdown();
```

5.4. Secondary index

5.4.1. Usage notes

Secondary indexes can be classified into global secondary indexes and local secondary indexes. This topic describes the terms of secondary indexes, and the differences between secondary index types. This topic also describes the limits and precautions of which you need to take note when you use secondary indexes.

Background information

The secondary index feature allows you to create one or more index tables for a data table. Then, you can query data based on the primary key columns of the index tables instead of the data table. This improves query efficiency.

Tablestore provides global secondary indexes and local secondary indexes to meet your requirements, such as strong query consistency.

Differences between secondary indexes types

Secondary indexes can be classified into global secondary indexes and local secondary indexes. You can use secondary indexes based on your business requirements. The following table describes the features of global secondary indexes and local secondary indexes to compare the differences between the two secondary index types.

Туре	Feature
Global secondary index	 Tablestore automatically synchronizes data from the indexed columns and primary key columns of a data table for which an index table is created to the index table in asynchronous mode. The synchronization latency is within a few milliseconds. The first primary key column of the index table can be a primary key column or a predefined column of the data table.
Local secondary index	 Tablestore automatically synchronizes data from the indexed columns and primary key columns of a data table for which an index table is created to the index table in synchronous mode. After data is written to the data table, you can immediately query the data from the index table. The first primary key column of the index table must be the first primary key column of the data table.

Terms

Term	Description
Index table	A table that is created based on the indexed columns of a data table. The data in an index table is read-only.

Predefined column	A non-primary key column that is predefined when you create a data table. A predefined column can be used as an attribute column of an index table. You need to specify the data type of a predefined column when you add the predefined column. Note Tablestore uses a schema-free model. You can write different attribute columns to a row without the need to specify the attribute columns in the table schema.
Single-column index	An index that is created on a single column.
Combined index	An index that is created on multiple columns. For example, a combined index can have index key columns 1 and 2.
Indexed attribute column	A predefined column that is mapped from a data table to an index table as a non-primary key column of the index table.
Autocomplete	The feature that allows the system to automatically add the primary key columns of the data table that are not specified as index key columns to an index table as the primary key columns of the index table.

Limits

- Each index table name must be unique in an instance.
- You can create up to five index tables for a data table. You cannot create index tables if the upper limit is reached.
- You can create up to 20 predefined columns for a data table. If the number of predefined columns exceeds the upper limit, the data table fails to be created.
- An index table can contain up to four index key columns, which are random combinations of the primary key columns and predefined columns of the data table. If the number of index key columns exceeds the upper limit, the index table fails to be created.
- An index table can contain up to eight attribute columns. If the number of attribute columns exceeds the upper limit, the index table fails to be created.
- You can set the data type of an index key column to STRING, INTEGER, or BINARY. The limits on index key columns are the same as those on primary key columns of the data table.
- If an index table is created based on multiple columns, the size limit on the columns is the same as that on primary key columns of the data table.
- If you specify a column of the STRING or BINARY type as an attribute column of an index table, the limits on the attribute column are the same as those on attribute columns of the data table.
- You cannot create an index table for a data table of which the time to live (TTL) is specified. If you want to create index tables on a data table of which the TTL is specified, use DingTalk to contact technical support.
- You cannot create an index table for a data table for which the Max Versions parameter is configured. If the Max Versions parameter is configured for
 a data table, index tables fail to be created for the data table. If an index table is created for the data table, you cannot configure the Max Versions
 parameter for the data table.
- You cannot specify the data versions if you write data to an indexed data table. Otherwise, data fails to be written to the data table.
- You cannot use the Stream feature for an index table.
- An indexed data table cannot contain repeated rows that have the same primary key value in a batch write operation. Otherwise, data fails to be written to the data table.

Precautions

The system automatically adds the primary key columns of the data table that are not specified as index key columns to an index table as the
primary key columns of the index table. When you scan data in an index table, you must specify the values of primary key columns. The values of
primary key columns range from negative infinity to positive infinity.

When you create an index table, you need to only specify the index key columns. The system automatically adds all primary key columns of the data table to the index table. For example, a data table contains primary key columns PK0 and PK1 and a predefined column Defined0.

If you use the global secondary index feature, you can create an index on columns based on your business requirements.

- If you create an index on the Defined0 column, Tablestore generates an index table whose primary key columns are Defined0, PK0, and PK1.
- If you create an index on the Defined0 and PK1 columns, Tablestore generates an index table whose primary key columns are Defined0, PK1, and PK0.
- If you create an index on the PK1 column, Tablestore generates an index table whose primary key columns are PK1 and PK0.

If you use the local secondary index feature, the first primary key column of an index table must be the same as the first primary key column of the data table for which the index table is created.

- If you create an index on the PK0 and Defined0 columns, Tablestore generates an index table whose primary key columns are PK0, Defined0, and PK1.
- If you create an index on the PK0, PK1, and Defined0 columns, Tablestore generates an index table whose primary key columns are PK0, PK1, and Defined0.
- If you create an index on the PK0 and PK1 columns, Tablestore generates an index table whose primary key columns are PK0 and PK1.

• You can specify predefined columns of a data table as attribute columns of the index table that is created for the data table based on your query modes and cost requirements.

After you specify a predefined column of a data table as an attribute column of the index table that is created for the data table, you can query the values in the predefined column from the index table. You do not need to query the data table. However, this increases storage costs. If a predefined column of the data table is not specified as an attribute column of the index table, you must query the values in the column from the data table.

- When you use the global secondary index feature, specify a column of a data table as the first primary key column of an index table based on your business requirements.
- If a column whose value is time or a date is the first primary key column of an index table, the update speed of the index table may decrease. Therefore, we recommend that you do not specify this type of column as the first primary key column of an index table.
- We recommend that you hash the column whose value is time or a date and create an index on the hashed column. If you need to use the hashing algorithm, use DingTalk to contact technical support.

 We recommend that you do not specify a column of low cardinality or a column that contains enumerated values as the first primary key column of an index table. For example, if you specify the gender column as the first primary key column of an index table, the horizontal scalability of the index table is limited. As a result, the write performance is compromised.

5.4.2. Use secondary indexes

The secondary index feature allows you to query data based on the primary key of a data table and the index key columns of the secondary index that is created for the data table. This improves query efficiency. When you create a secondary index, you can specify predefined columns of the data table for which you want to create the secondary index as index key columns or attribute columns of the secondary index. After you create a secondary index, you can use the secondary index to query data.

Usage notes

The first primary key column of an index table cannot be an auto-increment primary key column.

Prerequisites

A data table for which the Max Versions parameter is set to 1 is created. The Time to Live parameter of the data table must meet one of the following conditions:

- The Time to Live parameter of the data table is set to -1, which indicates that data in the data table never expires.
- The Time to Live parameter of the data table is set to a value other than -1, and the Allow Updates parameter is set to No.

```
⑦ Note
The time to live (TTL) of a secondary index is the same as that of the data table for which the secondary index is created.
```

Step 1: (Optional) Add predefined columns

If you create a secondary index for a data table and the data table does not contain predefined columns or the existing predefined columns do not meet your business requirements, you can add or delete predefined columns in the data table.

? Note

You can also add predefined columns when you create a data table. For more information, see Create a data table.

1. Go to the Manage Table page.

- i. Log on to the Tablestore console
- ii. On the **Overview** page, click the name of the instance that you want to manage.
- iii. In the Tables section of the Instance Details tab, click the name of the data table that you want to manage.
- 2. In the Pre-defined Column section of the Advanced Features module on the Basic Information tab, click Add Pre-Defined Column.
- 3. In the Add Pre-Defined Column dialog box, click Add. Specify a name for the predefined column and select a data type from the drop-down list.
- The name of a predefined column must be 1 to 255 bytes in length and can contain letters, digits, and underscores (_). The name must start with a letter or an underscore (_).
- Predefined columns support the following data types: STRING, INTEGER, BINARY, FLOAT, and BOOLEAN.
- If you want to use the predefined column as an index key column of a secondary index, select **STRING**, **INTEGER**, or **BINARY** from the drop-down list for the predefined column.

? Note

In the Add Pre-Defined Column dialog box, you can add multiple predefined columns. To delete a predefined column, click the icon next to the predefined column.

4. Click **OK**.

The predefined columns that you add are displayed in the Pre-defined Column section.

To delete a predefined column that you no longer need, find the predefined column and click the 🝵 icon in the Remove Pre-Defined Column column.

Step 2: Create a secondary index

- 1. Go to the Manage Table page.
- i. Log on to the Tablestore console.
- ii. On the **Overview** page, click the name of the instance that you want to manage.
- iii. In the Tables section of the Instance Details tab, click the name of the data table that you want to manage.
- 2. On the Indexes tab, click Create Secondary Index.
- 3. In the Create Index dialog box, configure the parameters.

i. The following table describes the parameters.

Parameter	Description
Index Type	The type of the index. The value of this parameter is set to Secondary Index and cannot be modified.
Instance Name	The name of the instance. You cannot modify the value of this parameter.
Table Name	The name of the data table for which you want to create the secondary index. You cannot modify the value of this parameter.
Index Name	The name of the secondary index that you want to create.
Index Type	 The type of the secondary index. Valid values: Global Secondary Index: Tablestore automatically synchronizes data from the indexed columns and primary key columns of the data table to the index table that you want to create in asynchronous mode. The first primary key column of the index table can be a primary key column or a predefined column of the data table. Local Secondary Index: Tablestore automatically synchronizes data from the indexed columns and primary key columns of the data table to the index table that you want to create in synchronous mode. After data is written to the data table, you can immediately query the data from the index table. The first primary key column of the index table must be the first primary key column of the data table.
Existing Data	Specifies whether to include existing data of the data table in the secondary index. Valid values: Include Existing Data Exclude Existing Data

ii. Select the primary key columns from the drop-down list in sequence based on the index type that you specify, and click Add Primary Key Column.

You can select only predefined columns of the STRING, INTEGER, or BINARY type as primary key columns for the secondary index.

iii. Select predefined columns as attribute columns for the secondary index and click Add Pre-defined Column.

4. Click **OK**.

The secondary index that you created is displayed in the Indexes section.

Step 3: Query data

You can query a single row of data or rows of data within the specified range.

Query data within the specified range

1. Go to the Manage Table page.

- i. Log on to the Tablestore console
- ii. On the **Overview** page, click the name of the instance that you want to manage.
- iii. In the Tables section of the Instance Details tab, click the name of the data table that you want to manage.
- 2. On the Indexes tab, find the index that you want to use to query data and click Query in the Actions column.
- 3. In the **Search** dialog box, specify the query conditions.
- i. Set the Modes parameter to Range Search.
- ii. By default, the system returns all columns. To return specific attribute columns, turn off **All Columns** and specify the attribute columns that you want to return. Separate multiple attribute columns with commas (,).
- iii. Configure the Start Primary Key Column and End Primary Key Column parameters.

? Note

- If you set the Modes parameter to Range Search, the range that is specified by the start value and the end value in the right primary key column takes effect only when the start value and the end value are the same within each leftmost primary key column. If the start value and the end value in a leftmost primary key column are different, the range that is specified by the start value and the end value in the right primary key column does not take effect.
- The range is a left-open and right-closed interval.
- iv. Use the default value of the Max Versions parameter.

The Max Versions parameter specifies the maximum number of versions of data in each attribute column that can be returned.

v. Specify the sorting order of the query results. You can select Forward Search or Backward Search.

4. Click **OK**.

The data that meets the query conditions is displayed on the lower part of the page.

Query a single row of data

1. Go to the Manage Table page.

- i. Log on to the Tablestore console
- ii. On the **Overview** page, click the name of the instance that you want to manage.
- iii. In the Tables section of the Instance Details tab, click the name of the data table that you want to manage.
- 2. On the Indexes tab, find the index that you want to use to query data and click Query in the Actions column.
- 3. In the **Search** dialog box, specify the query conditions.
 - i. Set the **Modes** parameter to **Get Row**.

- ii. By default, the system returns all columns. To return specific attribute columns, turn off **All Columns** and specify the attribute columns that you want to return. Separate multiple attribute columns with commas (,).
- iii. Configure the **Primary Key Value** parameter of the row that you want to query.
- The integrity and accuracy of the primary key values affect the query results
- iv. Use the default value of the Max Versions parameter. The Max Versions parameter specifies the maximum number of versions of data in each attribute column that can be returned.
- 4. Click **OK**.

If the row that you want to query is included in the index table, Tablestore returns the data of the row. If the row that you want to query is not included in the index table, no data is returned.

5.4.3. Common scenarios

The secondary index feature allows you to create an index on the specified columns. Data in the generated index table is sorted based on the specified index key columns. All data written to the data table is automatically synchronized to the index table. After you write data to the data table, you can query the data from the index table that is created for the data table. This improves the query efficiency.

Global secondary index

You can use the global secondary index feature to perform various queries on a data table of phone calls.

The following table describes the information about the data table. When a phone call is complete, the information about the phone call is recorded in the data table.

- The CellNumber and StartTime columns are used as the primary key columns of the data table. Each value in the CellNumber column indicates a caller, and each value in the StartTime column indicates the start time of a phone call.
- The CalledNumber, Duration, and BaseStationNumber columns are used as the predefined columns of the data table. Each value in the CalledNumber column indicates the number of a call recipient, each value in the Duration parameter indicates the duration of a phone call, and each value in the BaseStationNumber column indicates a base station number.

CellNumber	StartTime (UNIX timestamp)	CalledNumber	Duration	BaseStationNumber
123456	1532574644	654321	60	1
234567	1532574714	765432	10	1
234567	1532574734	123456	20	3
345678	1532574795	123456	5	2
345678	1532574861	123456	100	2
456789	1532584054	345678	200	3

You can create an index table on the CalledNumber and BaseStationNumber columns to perform various queries.

You can use the global secondary index feature to meet the following query requirements:

• Query the rows in which the value of the CellNumber column is 234567.

Tablestore sorts rows in a data table based on their primary keys and provides the getRange operation. If you call the getRange operation to query the rows in which the value of the CellNumber column is 234567, set both the maximum and minimum values of the CellNumber column to 234567, and set the minimum value and maximum value of the StartTime column to 0 and INT_MAX. This way, the data table is scanned to return the rows that meet the query condition.

private static void getRangeFromMainTable(SyncClient client, long cellNumber){
<pre>RangeRowQueryCriteria rangeRowQueryCriteria = new RangeRowQueryCriteria(TABLE_NAME);</pre>
// Construct the primary key.
PrimaryKeyBuilder startPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY KEY NAME 1, PrimaryKeyValue.fromLong(cellNumber));
startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY KEY NAME 2, PrimaryKeyValue.fromLong(0));
<pre>rangeRowQueryCriteria.setInclusiveStartPrimaryKey(startPrimaryKeyBuilder.build());</pre>
// Construct the primary key.
PrimaryKeyBuilder endPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.fromLong(cellNumber));
endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.INF_MAX);
<pre>rangeRowQueryCriteria.setExclusiveEndPrimaryKey(endPrimaryKeyBuilder.build());</pre>
<pre>rangeRowQueryCriteria.setMaxVersions(1);</pre>
<pre>String strNum = String.format("%d", cellNumber);</pre>
System.out.println("The cell number" + strNum + "makes the following calls:");
while (true) {
GetRangeResponse getRangeResponse = client.getRange(new GetRangeRequest(rangeRowQueryCriteria));
<pre>for (Row row : getRangeResponse.getRows()) {</pre>
System.out.println(row);
}
// If the value of the nextStartPrimaryKey parameter is not null, continue to read data.
if (getRangeResponse.getNextStartPrimaryKey() != null) {
rangeRowQueryCriteria.setInclusiveStartPrimaryKey(getRangeResponse.getNextStartPrimaryKey());
} else {
break;
}
}

• Query the rows in which the value of the CalledNumber column is 123456.

compromised. In this case, you can create an index table named IndexOnBeCalledNumber on the CalledNumber column. The CalledNumber column is used as the primary key column of the index table. Therefore, you can call the getRange operation to scan the index table to obtain the rows that meet the query condition.

The following table describes the information about the IndexOnBeCalledNumber index table.

? Note

Tablestore automatically adds the primary key columns of the data table that are not specified as index key columns to the index table. The primary key columns of the data table and the index key column are used as the primary key columns of the index table. Therefore, the index table contains three primary key columns.

РКО	РК1	РК2
CalledNumber	CellNumber	StartTime
123456	234567	1532574734
123456	345678	1532574795
123456	345678	1532574861
654321	123456	1532574644
765432	234567	1532574714
345678	456789	1532584054

private static void getRangeFromIndexTable(SyncClient client, long cellNumber) {
RangeRowQueryCriteria rangeRowQueryCriteria = new RangeRowQueryCriteria(INDEX0_NAME);
// Construct the primary key.
PrimaryKeyBuilder startPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
startPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_1, PrimaryKeyValue.fromLong(cellNumber));
startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MIN);
startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.INF_MIN);
<pre>rangeRowQueryCriteria.setInclusiveStartPrimaryKey(startPrimaryKeyBuilder.build());</pre>
// Construct the primary key.
<pre>PrimaryKeyBuilder endPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();</pre>
endPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_1, PrimaryKeyValue.fromLong(cellNumber));
endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MAX);
endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.INF_MAX);
rangeRowQueryCriteria.setExclusiveEndPrimaryKey(endPrimaryKeyBuilder.build());
<pre>rangeRowQueryCriteria.setMaxVersions(1);</pre>
<pre>String strNum = String.format("%d", cellNumber);</pre>
System.out.println("The cell number" + strNum + "is called by the following numbers:");
while (true) {
GetRangeResponse getRangeResponse = client.getRange(new GetRangeRequest(rangeRowQueryCriteria));
<pre>for (Row row : getRangeResponse.getRows()) {</pre>
System.out.println(row);
}
// if the Value of the nextstartPrimaryKey parameter is not hull, continue to read data.
if (getkangekesponse.getwextstattPrimaryKey() := null) {
<pre>Langerowguerycriteria.setinclusivestartPrimarykey(getkangekesponse.getwextStartPrimarykey());)</pre>
Diedk;
3

• Query the rows in which the value of the BaseStationNumber column is 002 and the value of the StartTime column is 1532574740.

This query is similar to the preceding queries except that two query conditions are specified in this query. Therefore, you can create an index table named IndexOnBaseStation1 on the BaseStationNumber and StartTime columns. Then, you can query data from the IndexOnBaseStation1 index table to obtain the rows that meet the query conditions.

The following table describes the information about the IndexOnBaseStation1 index table.

РКО	РК1	РК2
BaseStationNumber	StartTime	CellNumber
001	1532574644	123456
001	1532574714	234567
002	1532574795	345678
002	1532574861	345678
003	1532574734	234567
003	1532584054	456789

private static void getRangeFromIndexTable(SyncClient client, long baseStationNumber,
RangeRowQueryCriteria rangeRowQueryCriteria = new RangeRowQueryCriteria(INDEX1_NAME);
<pre>// Construct the primary key. PrimaryKeyBuilder startPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder(); startPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber)); startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(startTime)); startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MIN); rangeRowQueryCriteria.setInclusiveStartPrimaryKey(startPrimaryKeyBuilder.build());</pre>
<pre>// Construct the primary key. PrimaryKeyBuilder endPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder(); endPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber)); endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.INF_MAX); endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MAX); rangeRowQueryCriteria.setExclusiveEndPrimaryKey(endPrimaryKeyBuilder.build());</pre>
<pre>rangeRowQueryCriteria.setMaxVersions(1);</pre>
<pre>String strBaseStationNum = String.format("%d", baseStationNumber); String strStartTime = String.format("%d", startTime); System.out.println("All called numbers forwarded by the base station" + strBaseStationNum + "that start from" + strStartTime + "are listed:"); while (true) { GetRangeResponse getRangeResponse = client.getRange(new GetRangeRequest(rangeRowQueryCriteria)); for (December 1);</pre>
System.out.println(row);
<pre>// If the value of the nextStartPrimaryKey parameter is not null, continue to read data. if (getRangeResponse.getNextStartPrimaryKey() != null) { rangeRowQueryCriteria.setInclusiveStartPrimaryKey(getRangeResponse.getNextStartPrimaryKey()); } else { break; } }</pre>
}

• Query the rows in which the value of the BaseStationNumber column is 003 and the value of the StartTime column ranges from 1532574861 to 1532584054 and return only the values of the Duration column.

This query specifies the conditions based on the values of the BaseStationNumber and StartTime columns and returns only the values of the Duration column. You can query data from the IndexOnBaseStation1 index table that is used in the preceding query to obtain the primary key of the rows that meet the query conditions. Then, query data from the data table based on the primary key to obtain the values of the Duration column.

priva	ate static void getRowFromIndexAndMainTable(SyncClient client, long baseStationNumber, long startTime, long endTime, String colName) {
1	RangeRowQueryCriteria rangeRowQueryCriteria = new RangeRowQueryCriteria(INDEX1_NAME);
, 1 2 3 3 3 3	<pre>// Construct the primary key. PrimaryKeyBuilder startPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder(); startPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber)); startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(startTime)); startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MIN); rangeRowQueryCriteria.setInclusiveStartPrimaryKey(startPrimaryKeyBuilder.build());</pre>
- - - - - - - - - - - - - - - - - - -	<pre>// Construct the primary key. PrimaryKeyBuilder endPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder(); endPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber)); endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(endTime)); endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MAX); rangeRowQueryCriteria.setExclusiveEndPrimaryKey(endPrimaryKeyBuilder.build());</pre>
:	<pre>rangeRowQueryCriteria.setMaxVersions(1);</pre>
:	String strBaseStationNum = String.format("%d", baseStationNumber); String strStartTime = String.format("%d", startTime); String strEndTime = String.format("%d", endTime);
: : + "ii	<pre>System.out.println("The duration of calls forwarded by the base station" + strBaseStationNum + "from" + strStartTime + "to" + strEndTime s listed:"); while (true) { GetRangeResponse getRangeResponse.getRows()) { PrimaryKey curIndexPrimaryKey = row.getPrimaryKey(); PrimaryKeyColumn mainCalledNumber = curIndexPrimaryKey.getPrimaryKeyColumn(PRIMARY_KEY_NAME_1); PrimaryKeyColumn callStartTime = curIndexPrimaryKey.getPrimaryKeyColumn(PRIMARY_KEY_NAME_1); PrimaryKeyColumn callStartTime = curIndexPrimaryKey.getPrimaryKeyColumn(PRIMARY_KEY_NAME_2); PrimaryKeyBuilder mainTablePKBuilder = PrimaryKeyJuder.createPrimaryKeyBuilder(); mainTablePKBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, callStartTime.getValue()); primaryKey mainTablePKBuilder.build(); // Construct the primary key of the data table. // Query data from the data table. SingleRowQueryCriteria criteria = new SingleRowQueryCriteria(TABLE_NAME, mainTablePK); criteria.addColumnsToGet(colName); // Read the values of the Duration column from the data table. // Set the MaxVersions parameter to 1 to read the latest version of data. criteria.setMaxVersions(1); GetERowResponse getERowResponse.getRow(); } ? } } ? } ? } ? ? ? ? ? ? ? ? ? ?</pre>
	<pre>System.out.println(mainTableRow); }</pre>
	<pre>// If the value of the nextStartPrimaryKey parameter is not null, continue to read data. if (getRangeResponse.getNextStartPrimaryKey() != null) { rangeRowQueryCriteria.setInclusiveStartPrimaryKey(getRangeResponse.getNextStartPrimaryKey()); } else { break; }</pre>
	}

}

To improve query efficiency, you can create an index table named IndexOnBaseStation2 on the BaseStationNumber and StartTime columns and specify the Duration column as an attribute column of the index table. Then, query data from the IndexOnBaseStation2 index table to obtain the rows that met the query conditions.

The following table describes the information about the IndexOnBaseStation2 index table.

РКО	PK1	PK2	Defined0
BaseStationNumber	StartTime	CellNumber	Duration
001	1532574644	123456	60
001	1532574714	234567	10
002	1532574795	345678	5
002	1532574861	345678	100
003	1532574734	234567	20

003	1532584054	456789	200
<pre>private static void getRangeFromIndexTable(SyncClient client,</pre>			
RangeRowQueryCriteria rangeRowQueryCriteria = new RangeRowQueryCriteria (INDEX2_NAME); // Construct the primary key. PrimaryKeyBuilder startPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder(); startPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber)); startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(startTime)); startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MIN); rangeRowQueryCriteria.setLnclusiveStartPrimaryKey(startPrimaryKeyBuilder.build());			
<pre>// Construct the primary key. PrimaryKeyBuilder endPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder(); endPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber)); endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(endTime)); endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MAX); rangeRowQueryCriteria.setExclusiveEndPrimaryKey(endPrimaryKeyBuilder.build());</pre>			
<pre>// Specify the name of the column that you want to read. rangeRowQueryCriteria.addColumnsToGet(colName);</pre>			
String strBaseStationNum = String.format("%d", baseStationNumber); String strStartTime = String.format("%d", startTime); String strEndTime = String.format("%d", endTime);			
<pre>System.out.println("The duration of calls forwarded by the base station" + strBaseStationNum + "from" + strStartTime + "to" + strEndTime + "is listed:"); while (true) { GetRangeResponse getRangeResponse = client.getRange(new GetRangeRequest(rangeRowQueryCriteria)); for (Row row : getRangeResponse.getRows()) { System.out.println(row); } }</pre>			
<pre>// If the value of the nextStartPrimaryKey parameter is not null, continue to read data. if (getRangeResponse.getNextStartPrimaryKey() != null) { rangeRowQueryCriteria.setInclusiveStartPrimaryKey(getRangeResponse.getNextStartPrimaryKey()); } else { break; } }</pre>			

If you do not specify the Duration column as an attribute column of the index table, you must query data from the index table to obtain the primary key of the rows that meet the query conditions, and then query data from the data table based on the primary key to obtain the values of the Duration column. If you specify the Duration column as an attribute column of the index table, the data of the Duration column is stored in both the data table and the index table. This increases storage usage.

• Query the total, average, maximum, and minimum call duration of all phone calls forwarded by the 003 base station. The start time of the phone calls ranges from 1532574861 to 1532584054.

In this query, you want to obtain the statistics for the duration of all phone calls instead of the duration of each phone call. You can use the same query method that is used in the preceding query. Then, you can perform aggregation on the Duration column to obtain the results.

Local secondary index

You can use the local secondary index feature to perform various queries on a data table of phone calls.

The following table describes the information about the data table. When a phone call is complete, the information about the phone call is recorded in the data table.

- The CellNumber and StartTime columns are used as the primary key columns of the data table. Each value in the CellNumber column indicates a caller, and each value in the StartTime column indicates the start time of a phone call.
- The CalledNumber, Duration, and BaseStationNumber columns are used as the predefined columns of the data table. Each value in the CalledNumber column indicates the number of a call recipient, each value in the Duration parameter indicates the duration of a phone call, and each value in the BaseStationNumber column indicates a base station number.

CellNumber	StartTime (UNIX timestamp)	CalledNumber	Duration	BaseStationNumber
123456	1532574644	654321	60	1
123456	1532574704	236789	60	1
234567	1532574714	765432	10	1
234567	1532574734	123456	20	3

User Guide Tablestore

345678	1532574795	123456	5	2
345678	1532574861	123456	100	2
456789	1532584054	345678	200	3
456789	1532585054	123456	200	3
456789	1532586054	234567	200	3
456789	1532587054	123456	200	3

You can use the local secondary index feature to create an index table named LocalIndexOnBeCalledNumber on the CalledNumber column. Then, you can query the records of phone calls between called numbers and caller numbers.

The following table describes the information about the LocalIndexOnBeCalledNumber index table.

? Note

Tablestore automatically adds the primary key columns of the data table that are not specified as index key columns to the index table. The primary key columns of the data table and the index key column are used as the primary key columns of the index table. Therefore, the index table contains three primary key columns.

РКО	Defined0	PK1	Defined1	Defined2
CellNumber	CalledNumber	StartTime (UNIX timestamp)	Duration	BaseStationNumber
123456	236789	1532574704	60	1
123456	654321	1532574644	60	1
234567	123456	1532574734	20	3
234567	765432	1532574714	10	1
345678	123456	1532574795	5	2
345678	123456	1532574861	100	2
456789	123456	1532585054	200	3
456789	123456	1532587054	200	3
456789	234567	1532586054	200	3
456789	345678	1532584054	200	3

To query all the records from the caller number 456789 to the called number 123456, you need to only call the getRange operation by specifying the following settings: Set the maximum and minimum values of the PK0 column to 456789, the maximum and minimum values of the Defined0 column to 123456, and the minimum value and maximum value of the PK1 column to 0 and INT_MAX. This way, the data table is scanned to return the rows that meet the query condition. Sample code:

private static void getRangeFromMainTable(SyncClient client, long cellNumber, long calledNumber) { RangeRowQueryCriteria rangeRowQueryCriteria = new RangeRowQueryCriteria (TABLE NAME); // Construct the start primary key. PrimaryKeyBuilder startPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder(); startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_0, PrimaryKeyValue.fromLong(cellNumber)); startPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED COL NAME 0, PrimaryKeyValue.fromLong(calledNumber)); startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.fromLong(0)); rangeRowQueryCriteria.setInclusiveStartPrimaryKey(startPrimaryKeyBuilder.build()); // Construct the end primary key. PrimaryKeyBuilder endPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder(); endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_0, PrimaryKeyValue.fromLong(cellNumber)); endPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_0, PrimaryKeyValue.fromLong(calledNumber)); endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MAX); rangeRowQueryCriteria.setExclusiveEndPrimaryKey(endPrimaryKeyBuilder.build()); rangeRowQueryCriteria.setMaxVersions(1); String strNum = String.format("%d", cellNumber); String strCalledNum = String.format("%d", calledNumber); System.out.println("All records of phone calls between the caller number" + strNum + "and the called number" +strCalledNum+ "are listed:"); while (true) { GetRangeResponse getRangeResponse = client.getRange(new GetRangeRequest(rangeRowQueryCriteria)); for (Row row : getRangeResponse.getRows()) { System.out.println(row); // If the value of the nextStartPrimaryKey parameter is not null, continue to read data. if (getRangeResponse.getNextStartPrimaryKey() != null) { rangeRowQueryCriteria.setInclusiveStartPrimaryKey(getRangeResponse.getNextStartPrimaryKey()); } else { break; } } }

5.5. Search index

5.5.1. Overview

The search index feature provides multiple efficient index schemas to help you process complex queries in big data scenarios.

Purposes

Data tables in Tablestore use distributed NoSQL data structures. Data such as monitoring data and log data can be stored, read, and written at a large scale.

In addition to queries based on primary keys including single-row read and range read, Tablestore provides the search index feature to meet your requirements for complex queries. These queries include Boolean query and queries based on non-primary key columns.

The search index feature is implemented by using inverted indexes and column stores. This feature provides query methods to solve problems in complex big data scenarios. The query methods include queries based on non-primary key columns, full-text search, prefix query, fuzzy query, Boolean query, nested query, and geo query. Aggregation can be implemented by using max, min, count, sum, avg, distinct_count, group_by, percentile statistics, and query by histogram.

Differences among indexes

Aside from queries based on primary keys in data tables, Tablestore provides two index schemas for accelerated queries: secondary index and search index. The following table describes the differences among the three types of indexes.

Index type	Description	Scenario
Primary key of a data table	A data table is similar to a large map. Data tables support queries based only on primary keys.	You can specify a complete primary key or the prefix of a primary key.
Secondary index	You can create one or more index tables and perform queries by using the primary key columns of the index tables.	You can define the required columns in advance. Therefore, only a small number of columns are queried. You can also specify a complete primary key or the prefix of a primary key.
Search index	The search index feature uses inverted indexes, Bkd-trees, and column stores for various query scenarios.	The following scenarios do not support the use of a primary key of a data table and the secondary index feature: • Query based on non-primary key columns • Boolean query • Query by using operators such as AND, OR, and NOT • Full-text search • Geo query • Prefix query • Fuzzy query • Nested query • Nested query • Aggregation that is implemented by using max, min, count, sum, avg, distinct_count, group_by, percentile statistics, and query by histogram

Compared with indexes of conventional database services such as MySQL, the search index feature is not subject to the leftmost matching principle. Therefore, the search index feature can be used in more scenarios. In most cases, only one search index is required for a data table. For example, a data table about student information may contain the student name, ID, gender, grade, class, and home address columns. When you create a search index, you can add these columns to the search index. When you use the search index, you can specify a combination of conditions. Examples: students named Tom in Grade Three, male students who live one kilometer away from their school, and students in Class Two, Grade Three who live in the specified residential community.

API operations

The search index feature provides the Search and ParallelScan API operations. The Search API operation is used for general queries. The ParallelScan API operation is used for data export.

Most features that are provided by the two API operations are the same. However, to improve the performance and throughput, the ParallelScan API operation does not provide some features of the Search API operation. The following table compares the two API operations.

Operation	Description
Search	 Supports all features of search indexes. Query: query based on non-primary key columns, full-text search, prefix query, fuzzy query, Boolean query, nested query, and geo query. Collapse (distinct) Sorting Aggregation Total number of rows
ComputeSplits+ParallelScan	 Exports data in parallel. The query feature of search indexes is supported. However, analysis features such as sorting and aggregation are not supported. The query speed of a ParallelScan request that includes a parallel scan task is five times faster than the query speed of a Search request. Query: query based on non-primary key columns, full-text search, prefix query, fuzzy query, Boolean query, nested query, and geo query. Multiple parallel scan tasks included in one ParallelScan request

Usage notes

() Important Predefined columns are not required when you use a search index.

Index synchronization

After a search index is created for a data table, data is written to the data table first. After the data is written to the data table, a success message is returned. At the same time, another asynchronous thread reads the newly written data from the data table and writes the data to the search index. The write performance of Tablestore is not affected when data is being asynchronously synchronized from a data table to a search index.

In most cases, the latency generated when data is synchronized to a search index is within 3 seconds. You can view the latency in real time in the Tablestore console.

- Time to live (TTL)
- If the UpdateRow operation is disabled for a data table, you can use the TTL feature of the search index that is created for the data table.
- If you want to retain the data only for a period of time and the time field does not need to be updated, you can implement the TTL feature by splitting a data table into several time-specific data tables.
 - This solution is implemented based on the following principles and rules, and has the following benefits:

• Principle: Split a data table based on fixed periods of time, such as day, week, month, or year. Then, you can create a search index for each table. This way, tables for the specified periods of time are retained.

For example, if you want to retain data for six months, you can store the data for each month in a data table (such as table_1, table_2, table_3, table_4, table_5, and table_6) and create a search index for each data table. Each data table and search index store the data only of a single month. Then, you need to only delete data tables that are retained for more than six months to implement the same feature as TTL.

When you query data by using a search index, you need to only query one table if data that meets the time range requirement is in that table. If data that meets the time range requirement is included in multiple tables, you need to query all these tables and then combine the query results.

- Rule: The size of a single index cannot exceed 50 billion rows. The search index feature provides the optimal query performance if the size of a single index does not exceed 20 billion rows.
- Benefits:
- You can adjust the data storage duration based on the number of data tables retained.
- Query performance is directly proportional to data volumes. After a data table is split into multiple data tables, the data volume of each data table has an upper limit. This helps ensure better query performance and avoid high query latency or timeouts.
- Max versions

You cannot create a search index for a data table for which you have specified the max versions parameter.

You can specify the timestamp when you write data to a column that allows only a single version. If you first write data with a greater version number and then write data with a smaller version number, the data with the greater version number may be overwritten by data with the smaller version number.

The results of the Search and ParallelScan requests may not include the timestamp attribute.

Features

Search indexes can solve complex query problems in big data scenarios. Other systems such as databases and search engines can also solve data query problems. The following figure shows the differences between Tablestore, databases, and search engines.

Tablestore can provide all features of databases and search engines except for JOIN operations, transactions, and relevance of search results. Tablestore also has high data reliability of databases and supports advanced queries of search engines. Therefore, Tablestore can be used to replace the common architecture that consists of databases and search engines. If you do not need JOIN operations, transactions, and relevance of search results, we recommend that you use the search index feature of Tablestore.



5.5.2. Features

This topic describes the core features of search indexes and the equivalent SQL syntax of some features of search indexes.

Core features

• Query based on non-primary key columns

In some scenarios, queries based only on primary key columns or the prefixes of primary key columns cannot meet the query requirements. If you want to query a row based on a column, you can create a search index on the column and perform queries based on non-primary key columns.

Boolean query

A Boolean query is suitable to search for orders. A table that stores the data of the orders may contain dozens of columns. It may be difficult to determine how to combine the columns required for queries when you create a table. Even if you have determined the columns required for queries, hundreds of combinations may be available. If you use a relational database service, you may have to create hundreds of indexes. In addition, if a combination is not specified in advance, you cannot query data if you have not created the corresponding index.

However, you can use Tablestore to create a search index that includes the required column names. You can combine the columns to query data based on your business requirements. Search indexes also support logical operators such as AND, OR, and NOT.

· Geo query

As mobile devices gain popularity, geographical location data becomes increasingly important. The geographical location data is used in many apps such as WeChat, Weibo, food delivery apps, sports apps, and Internet of Vehicles (IoV) apps. To use the geographical location data, these apps must support the geo query feature.

Search indexes support the following query features based on geographical location data:

- Near: queries points within the specified radius based on a central point, such as the People Nearby feature in WeChat.
- Within: queries points within the specified rectangular or polygonal area.

Tablestore allows you to use these features to query geographical location data without using other database services or search engines.

Full-text search

Search indexes can tokenize data to perform full-text searches. Unlike search engines, Tablestore returns only BM25-based results. Tablestore does not return custom relevance results in the response. If you need to query relevance results, we recommend that you use search engines.

Search indexes support the following tokenization methods: single-word tokenization, delimiter tokenization, minimum semantic unit-based tokenization, and fuzzy tokenization.

• Fuzzy query

Search indexes support wildcard queries. This feature is similar to the LIKE operator in relational databases. You can specify characters and wildcards such as question marks (?) or asterisks (*) to implement the similar feature as the LIKE operator.

· Prefix query

Search indexes support prefix queries. This feature is applicable to all natural languages. For example, if you perform a prefix query based on the prefix apple, results such as apple6s and applexr may be returned.

Nested query

In addition to a flat structure, online data such as tagged pictures has complex and multilayered structures. For example, a database stores a large number of pictures, and each picture has multiple elements such as houses, cars, and people. Each element in a picture has a unique weight score. The score is evaluated based on the size and position of an element in a picture. Therefore, each picture has multiple tags. Each tag has a name and a weight score. You can use nested queries to query data based on the data tags.

The following sample code provides examples of the tags in JSON format:

```
{
    "tags": [
    {
        "name": "car",
        "score": 0.78
    },
    {
        "name": "tree",
        "score": 0.24
    }
]
}
```

You can perform nested queries to store and query multilayered data. This helps model complex data.

Collapse (distinct)

Search indexes support the collapse (distinct) feature. This feature allows you to specify the highest frequency of occurrence of an attribute value to achieve high cardinality. For example, if you search for a laptop on an e-commerce platform, the first page may display only products of a single brand, which is not user-friendly. You can use the collapse (distinct) feature of Tablestore to resolve this issue.

• Sorting

In Tablestore, data is sorted in alphabetical order based on the primary key. To sort data based on other columns, use the sorting feature. Tablestore supports a variety of sorting methods, including ascending, descending, single-field, and multi-field sorting. By default, the returned results are sorted by primary key. The data in a search index is globally sorted.

Total number of rows
You can specify the number of rows that Tablestore returns for the current request when you use a search index to perform a query. If you do not specify the query conditions for the search index, Tablestore returns the total number of rows for which you have created indexes. If you stop writing new data to a table and you have created indexes for all data in the table, Tablestore returns the total number of rows in the table. This feature applies to data verification and data-based operations.

Aggregation

Search indexes support the aggregation feature to allow you to obtain the minimum value, maximum value, sum, average, count and distinct count of rows, and percentile statistics. You can also perform aggregation operations to group results or perform queries by histogram. This helps meet the basic aggregation requirements in lightweight analysis scenarios.

SQL

Tablestore does not support SQL statements or operators. However, Tablestore provides equivalent search index features for most of SQL syntax. The following table describes the equivalent SQL syntax of some features of search indexes.

SQL syntax	Search index feature
Show	DescribeSearchIndex operation
Select	ColumnsToGet parameter
From	index name parameter Important Single-column indexes are supported. Multi-column indexes are not supported.
Where	A variety of query methods such as term queries
Order by	sort parameter
Limit	limit parameter
Delete	Perform a query and call the DeleteRow operation
Like	Wildcard query
And	operator = and parameter
Or	operator = or parameter
Not	Boolean query with mustNotQueries specified
Between	Range query
Null	Exists query
In	Terms query
Min	Obtain the minimum value by using the aggregation feature
Max	Obtain the maximum value by using the aggregation feature
Avg	Obtain the average value by using the aggregation feature
Count	Obtain the count of rows by using the aggregation feature
Count(distinct)	Obtain the distinct count of rows by using the aggregation feature
Sum	Obtain the sum by using the aggregation feature
Group By	GroupBy

5.5.3. Limits

This topic describes the limits of search indexes.

Mapping

Cloud Defined Storage

Item	Maximum value	Description
Number of indexed fields	500	The number of fields that can be indexed.
Array length	256	The maximum number of elements in an array.
Number of fields for which the EnableSortAndAgg parameter is set to true	100	The number of fields that can be sorted and aggregated.
Number of layers that can be nested in a query	2	A maximum number of two layers can be nested in a query.
Number of child rows in a nested field	50	The maximum number of child rows in a nested field.
Number of nested fields	25	The number of child fields that can be nested.
Total length of values in all primary key columns	1,000 bytes	The total length of values in all primary key columns in each row can be up to 1,000 bytes.
Length of the value in a primary key column of the STRING type	1,000 bytes	To index a primary key column of the STRING type, the value of the column cannot exceed 1,000 bytes in length.
Length of the value in an attribute column of the STRING type if you want to index the column as KEYWORD	1 KB	An indexed field can be up to 1 KB in length. If the length of an indexed field exceeds 1 KB, the excess part is truncated. The field length in a data table does not conform to this limit.
Length of the value in an attribute column of the STRING type if you want to index the column as TEXT	256 КВ	An indexed field can be up to 256 KB in length. If the length of an indexed field exceeds 256 KB, the excess part is truncated. The field length in a data table does not conform to this limit.
Length of a query string for a wildcard query	32	The query string can be up to 32 characters in length.
Length of a query string for a prefix query	1,000 bytes	The query string can be up to 1,000 characters in length.

Search

Category	Item	Maximum value	Description
	offset+limit	10000	To increase the number of returned rows, configure the next_token parameter.
	limit	100	 If you call the Search operation to query data of the specified column, the value of the limit parameter can be up to 1000 if the data is contained in search indexes. This means that up to 1,000 rows can be returned per request. To increase the upper limit, contact Alibaba Cloud technical support.
General limits	timeout	10s	None.
	CU	Unlimited.	None.
QPS Number of query methods specified in a Search call	QPS	100,000	 The upper limit for lightweight transaction processing is 100,000 queries per second (QPS). To increase the upper limit, contact Alibaba Cloud technical support.
	Number of query methods that are specified in a Search call	1024	If complex nested queries are specified in a Search call, query performance is compromised. We recommend that you simplify the query.
	Number of terms that are specified in a terms query	1024	If a large number of terms are specified in a terms query, the query efficiency is compromised. We recommend that you simplify the query or perform multiple queries.
	Number of Aggregation operations at the same level	5	The number of Aggregation operations is recalculated each time you add a new aggregation clause to a sub group.

User Guide Tablestore

	Number of GroupBy operations at the same level	5	The number of GroupBy operations is recalculated each time you add a new GroupBy clause to a sub group.
	Number of layers that can be nested in a group	3	The root group is calculated as a nested layer.
Aggregation	Number of filters specified in a GroupByFilter operation	10	None.
	Number of groups that are returned by each GroupByField operation	2,000	None.
	Number of ranges that are specified in a GroupByRange operation	100	None.
	Number of ranges that are specified in a GroupByGeodistance operation	10	None.

ParallelScan

Category	Item	Description
	offset+limit	If you use the parallel scan feature, you cannot configure the offset and limit parameters. The returned results are displayed in chronological order.
	limit	Maximum value: 1000.
General limits	CU	Unlimited.
	QPS	Unlimited.
	Maximum number of parallel scan tasks in a ParallelScan call	The value of the MaxParallel parameter. You can call the ComputeSplits operation to obtain the value of the parameter.
	Maximum number of parallel scan tasks	The maximum number of concurrent parallel scan tasks is 10. Parallel scan tasks that have the same session ID and same value of the ScanQuery parameter are considered one task.

Index

Item	Maximum value	Description
Rate	50,000 rows/s	 The first time when data is written to a data table or when a large volume of data is written in a short period of time, Tablestore balances loads within a few minutes. The maximum rate of indexing TEXT fields is 10,000 rows/s because this process consumes a large number of CPU resources for tokenization. To increase the upper limit, contact Alibaba Cloud technical support.
Synchronization latency	3s	 In most cases, the synchronization latency is within 3 seconds. It takes up to 1 minute to initialize a new index.
Number of rows in an index	100 billion	To increase the upper limit, contact Alibaba Cloud technical support.
Total size of data in an index	100 TB	To increase the upper limit, contact Alibaba Cloud technical support.

5.5.4. Data type mappings

This topic describes the mappings between field data types in data tables and search indexes. This topic also describes the additional attributes supported by different data types of fields.

The value of a field in a search index is the value of the field with the same name in the data table for which the search index is created. The data types of the two values must match. The following table describes the mappings between field data types in data tables and search indexes.

() Important

The data type of a field in the search index must match the data type of the field with the same name in the data table based on the mappings described in the following table. Otherwise, Tablestore discards the data as dirty data. Make sure that the values of the fields of the GEOPOINT and NESTED types comply with the formats described in the following table. If the format of data is invalid, Tablestore discards the data as dirty data. In this case, the data may be available in the data table but unavailable for queries in the search index.

Field data type in search indexes	Field data type in data tables	Description
LONG	INTEGER	A 64-bit long integer.
DOUBLE	DOUBLE	A 64-bit double-precision floating-point number.
BOOLEAN	BOOLEAN	A Boolean value.
KEYWORD	STRING	A string that cannot be tokenized.
TEXT	STRING	A string or text that can be tokenized.
DATE	INTEGER and STRING	Data of the DATE type. You can write the data in multiple date formats.
GEOPOINT	STRING	The coordinate pair of a geographical location, in the format of latitude,longitude . Valid values of the latitude: [-90,+90]. Valid values of the longitude: [-180,+180]. Example: 35.8,-45.91 .
NESTED	STRING	Data of the NESTED type. Example: [{"a": 1}, {"a": 3}].

The following table describes the additional attributes of fields in search indexes.

Attribute	Туре	Description
IsArray	BOOLEAN	Specifies whether the value of the column is an array. A value of true indicates that the value of the column is an array. The data written to the column must be a JSON array such as ["a", "b", "c"]. You do not need to specify the IsArray attribute for a column of the NESTED type because the value of the column is an array. You can perform all types of queries on data of the ARRAY type because arrays do not affect the query results.

The following table describes the combinations of data types and field attributes.

Data type	IsArray	isVirtualField
LONG	Supported	Supported
DOUBLE	Supported	Supported
BOOLEAN	Supported	Not supported
KEYWORD	Supported	Supported
ТЕХТ	Supported	Supported
DATE	Supported	Supported
GEOPOINT	Supported	Supported
NESTED	Fields of the NESTED type are arrays.	Not supported

5.5.5. Basic features

5.5.5.1. Search indexes

Search indexes use inverted indexes and column stores to address complex queries that involve a large amount of data. After you create a search index, you can use the search index to query data.

Prerequisites

A data table for which the Max Versions parameter is set to 1 is created. The Time to Live parameter of the data table must meet one of the following requirements:

• The Time to Live parameter of the data table is set to -1, which indicates that data in the data table never expires.

• The Time to Live parameter of the data table is set to a value other than -1, and update operations on the data table are disabled. For more information, see Create a data table.

Step 1: Create a search index

1. Go to the **Indexes** tab.

i. Log on to the Tablestore console

- ii. On the Overview page, click the name of the instance that you want to manage or click Manage Instance in the Actions column of the instance.
- iii. In the Tables section of the Instance Details tab, click the name of the data table for which you want to create a search index. On the Manage Table page, click Indexes. You can also click Indexes in the Actions column of the data table.

2. On the Indexes tab, click Create Search Index.

- 3. In the Create Index dialog box, configure the required parameters.
 - i. By default, the system generates a name for the search index. You can also specify a name for the search index.
 - ii. The default value of the Time to Live parameter is -1 for the search index. You can specify another value for the **Time to Live** parameter based on your business requirements.

The value of the Time to Live parameter must be -1 or a value that is greater than or equal to 86400. Unit: seconds. A value of -1 indicates that data in the search index never expires. A value of 86400 indicates one day. The time to live (TTL) value of the search index must be smaller than or equal to the TTL value of the data table for which the search index is created.

! Important

To use the TTL feature, you must set the Allow Updates parameter to No for the data table for which the search index is created.

iii. Configure the $\ensuremath{\textbf{Schema Generation Type}}$ parameter.

! Important

The values of the **Field Name** and **Field Type** parameters must match those of the data table. For more information about the mappings between the field types in data tables and those in search indexes, see **Data type mappings**.

- If you set the **Schema Generation Type** parameter to **Manual**, you must specify the field names, select field types, and specify whether to turn on Array.
- If you set the **Schema Generation Type** parameter to **Auto Generate**, the system automatically uses the primary key columns and attribute columns of the data table as index fields. You can select field types and specify whether to turn on Array based on your business requirements.

? Note

If you need to optimize the query performance, you can use virtual columns. For more information, see Virtual columns.

4. Click **OK**.

After the search index is created, find the search index in the Indexes section and click **Index Details** in the **Actions** column. In the Index Details dialog box, you can view the information about the search index, such as the metering information and index fields.

Step 2: Query data

1. Go to the **Indexes** tab.

- i. Log on to the Tablestore console
- ii. On the Overview page, click the name of the instance that you want to manage or click Manage Instance in the Actions column of the instance.
- iii. In the **Tables** section of the **Instance Details** tab, click the name of the data table in which you want to query data. On the Manage Table page, click Indexes. You can also click **Indexes** in the **Actions** column of the data table.
- 2. On the Indexes tab, find the search index that you want to use to query data and clickManage Data in the Actions column.
- 3. In the Search dialog box, specify the query conditions.
- i. By default, the system returns all columns. To return specific attribute columns, turn off **All Columns** and specify the attribute columns that you want to return. Separate multiple attribute columns with commas (,).

? Note

By default, the system returns all primary key columns of the data table.

ii. Select index fields from the drop-down list. Click **Add** each time you select an index field. Configure the Query Type and Value parameters. The following table describes the query types.

Query type	Description
Term query	This query uses full and exact matches to retrieve data from a data table. A term query is similar to a query based on a specific string. If a column of the TEXT type is queried and one of the tokens in a row exactly matches the specified keyword, the row meets the query conditions.
Range query	This query retrieves data within the specified range from a data table. If a column of the TEXT type is queried and one of the tokens in a row is within the specified range, the row meets the query conditions.
Prefix query	This query retrieves data that contains the specified prefix from a data table. If a column of the TEXT type is queried and one of the tokens in a row contains the specified prefix, the row meets the query conditions.
Match query	This query uses approximate matches to retrieve data from a data table. The keyword that you use for a match query and the column values are tokenized based on the analyzer that you specify. Then, a match query is performed based on the tokens. The OR logical operator is used to relate tokens. If the number of tokens in a row that match the tokens in the tokenized keyword reaches the minimum value that you specify, the row meets the query conditions.
Wildcard query	This query retrieves data from a data table if the data matches a string that contains one or more wildcard characters. You can use the asterisk (*) and question mark (?) wildcard characters in a string. The asterisk (*) matches a string of any length in, before, or after a term that you want to query. The question mark (?) matches a single character in a specific position.
Match phrase query	This query is similar to a match query. A row meets the query conditions only when the order and positions of the tokens in the row match the order and positions of the tokens that are contained in the tokenized keyword.
Exists query	An exists query is also called a NULL query or a NULL-value query. This query is used for sparse data to determine whether a column of a row exists. For example, you can query the rows in which the value of the address column is not empty. If a column does not exist in a row or the value of the column is an empty array ("[]"), the column does not exist in the row.
Terms query	This query is similar to a term query except that you can specify multiple keywords at the same time. If one of the tokens in a row matches one of the keywords, the row meets the query conditions.
Boolean query	This query retrieves data based on one or more subqueries. Tablestore returns the rows that meet the conditions of the one or more subqueries. Subquery conditions can be combined by using logical operators, such as AND, NOT, and OR. If subquery conditions are combined by using the AND or OR operator, you must specify multiple fields. If subquery conditions are combined by using the NOT operator, you need to only specify one field.
Nested query	This query retrieves data of nested fields.
Geo-distance query	You can specify a circular geographic area that consists of a central point and radius as a query condition. Tablestore returns the rows in which the value of the specified column falls within the circular geographic area.
Geo-bounding box query	You can specify a rectangular geographic area as a query condition. Tablestore returns the rows in which the value of the specified column falls within the rectangular geographic area.
Geo-polygon query	You can specify a polygon geographic area as a query condition. Tablestore returns the rows in which the value of the specified column falls within the polygon geographic area.

iii. By default, the sorting feature is disabled. To sort the query results based on index fields, turn on **Sort**, add the index fields based on which the query results are sorted, and specify a sorting method.

4. Click **OK**.

Data that meets the query conditions is displayed in the specified order on the **Indexes** tab.

5.5.5.2. Lifecycle management

The time to live (TTL) can be configured for search indexes. TTL is an attribute of search indexes that specifies the retention period of data in search indexes. When data in a search index is retained for a period of time that exceeds the TTL value, Tablestore automatically deletes the data to free up storage space and reduce costs.

Usage notes

• To use the TTL feature of a search index, you must prohibit the UpdateRow operation on the data table for which the search index is created due to the following reasons:

The TTL feature of data tables takes effect on attribute columns, and the TTL feature of search indexes takes effect on the entire rows. If the UpdateRow operation is performed on a data table, when the system clears data in the data table, the values of some fields are deleted and the values of some fields are retained in the data table. However, the entire rows in the search index that is created for the data table are not deleted. As a result, data in the data table and search index is inconsistent.

If the UpdateRow operation is required, check whether the UpdateRow operation can be changed to the PutRow operation.

- The TTL value of search indexes can be -1 or a positive int32 in seconds. The value of -1 indicates that data in the search index never expires and the maximum int32 value is equivalent to approximately 68 years.
- The TTL value of a search index is independent of and must be smaller than or equal to the TTL value of the data table for which the search index is created. If you need to change TTL values of search indexes and data tables for which the search indexes are created to smaller values, you must change the TTL values of the search indexes before you change the TTL values of the data tables.
- Tablestore automatically deletes expired data from search indexes every day. In some cases, you can still query expired data in search indexes. Tablestore automatically deletes the expired data in the next cycle.
- After you change the TTL values of data tables and search indexes, the system automatically deletes legacy expired data from the data tables and search indexes in the next cycle.

Procedure

You can configure the TTL of a search index by using the Tablestore console. To use the TTL feature of a search index, you must prohibit the UpdateRow operation on the data table for which the search index is created.

1. Prohibit the UpdateRow operation on a data table.

i. On the $\ensuremath{\textbf{Basic}}$ information tab of the data table, click $\ensuremath{\textbf{Modify}}$ $\ensuremath{\textbf{Attributes}}.$

ii. In the Modify Attributes dialog box, select No for Allow Updates and make sure that you are aware of the risks.

() Important After you prohibit the UpdateRow operation on a data table, all data update operations on the data table by using UpdateRow are prohibited.

iii. Click OK.

2. Specify the TTL for search indexes.

After the UpdateRow operation on a data table is prohibited, you can specify the TTL of a search index when you create the search index or modify the TTL of existing search indexes. The TTL value of a search index must be smaller than or equal to the TTL value of the data table for which the search index is created.

- Specify the TTL when you create a search index
- a. On the Indexes tab of the data table for which you want to create a search index and on which the UpdateRow operation is prohibited, click Create Search Index.
- b. In the Create Index dialog box, configure the Index Name, Time to Live, and Schema Generation Type parameters.

c. Click OK.

• Modify the TTL for an existing search index

a. On the **Indexes** tab of the data table on which the UpdateRow operation is prohibited, find the search index for which you want to modify the TTL and click **Index Details** in the **Actions** column.

- b. In the Index Details dialog box, click Modify.
- c. In the dialog box that appears, modify the Time to Live value and click Modify.
- d. Click OK.
- 3. The TTL of a data table is independent of the TTL of the search index that is created for the data table. If you want to use the TTL of a data table, configure the TTL for the data table.
 - i. In the Description section of the Basic Information tab of a data table, click Modify Attributes
- ii. In the Modify Attributes dialog box, configure the Time to Live parameter based on your business requirements.
- iii. Click OK

5.5.5.3. Types of date data

The search index feature can identify date type data in various formats. You can index dates stored as strings or integers to use in the search index. Queries made on dates take less time than queries made on strings in search indexes.

Accuracy and range

The maximum precision of date data is the nanosecond. The value range of dates is ["1970-01-01 00:00:00.000000000", "2262-04-11 23:47:16.854775807"].

Date formats

You can specify the data type in which dates are stored when you create search indexes. The following table describes the data types and the supported formats.

Data type	Supported format
INTEGER	 You can use a predefined format. The following predefined formats are provided: "epoch_second": indicates a timestamp in seconds. For example, a value of 1218197720 indicates 2008-08-08 20:15:20. "epoch_millis": indicates a timestamp in milliseconds. For example, a value of 1218197720123 indicates 2008-08-08 20:15:20.123. "epoch_micros": indicates a timestamp in microseconds. For example, a value of 1218197720123456 indicates 2008-08-08 20:15:20.123456. "epoch_nanos": indicates a timestamp in nanoseconds. For example, a value of 1218197720123456789 indicates 2008-08-08 20:15:20.123456789.

STRING	Custom formats are supported. The following list describes some commonly used date formats: yyyy-MM-dd HH:mm:ss.SSS yyyyMMdd HHmmss yyyy-MM-dd'T'HH:mm:ss.SSSX In the preceding formats, yyyy indicates the four-digit year, MM indicates the month, dd indicates the day, HH indicates the 24-hour clock, mm indicates the minute, ss indicates the second, SSS indicates the precision of the second, and X indicates the offset of the time zone. For more information, see the Custom date format section of this topic.

Custom date format

Character	Description	Example
У	Indicates the year.	 yyyy: 2008 yy: 08
М	Indicates the month.	• M: 7 • MM: 07
d	Indicates the day in the month.	• d: 8 • dd: 08
a	Indicates the time period of a day. In the 12-hour clock system, a day is divided into an ante meridiem (AM) and a post meridiem (PM), which indicates the time period before or post midday.	• a: AM • a: PM
К	Indicates the hour in AM or PM. Valid values: 0 to 11.	• K: 0 • KK: 00
н	Indicates the hour in a day in the 24-hour clock system. Value values: 0 to 23.	• H: 0 • HH: 00
m	Indicates the minute.	• m: 1 • mm: 01
s	Indicates the second.	• s: 1 • ss: 01
S	The precision of the second. Valid values: 0 to 999999999.	S: 3SSS: 234SSSSSSSS: 123456789
x	Indicates the offset of the time zone.	 X: +01;Z XX: +0130;Z XXX: +01:30;Z XXXX: +013015;Z XXXXX: +01:30:15;Z
x	Indicates the offset of the time zone.	 x: +01;+00 xx: +0130;+0000 xxx: +01:30;+00:00 xxxx: +01:3015;+0000 xxxxx: +01:30:15
	The custom qualifier.	You can use letters to define custom qualifiers. If you use a string as the qualifier, the string must be enclosed in a pair of single quotation marks ("). () Important Spaces and hyphens (-) do not need to be enclosed in single quotation marks (").
п	The escape character.	

Verify a date format

Before you use a date format to query data, we recommend that you verify whether the date format is valid.

User Guide Tablestore

• Create a search index that includes the date format to be verified, and perform a term query to verify the date format. The following sample code provides an example:

```
public void testDateFormat(SyncClient client, String tableName, String indexName) {
    // Create a search index.
    CreateSearchIndexRequest request = new CreateSearchIndexRequest();
   request.setTableName(tableName);
    request.setIndexName(indexName);
    IndexSchema indexSchema = new IndexSchema();
    indexSchema.setFieldSchemas(Arrays.asList(
       new FieldSchema("col_date", FieldType.DATE)
       .setIndex(true)
       .setEnableSortAndAgg(true)
        .setDateFormats(Arrays.asList("yyyy-MM-dd HH:mm:ss.SSS"))
   ));
    request.setIndexSchema(indexSchema);
   client.createSearchIndex(request);
    // Verify the date format. If no error is returned, the date format is valid.
   client.search(SearchRequest.newBuilder()
                 .tableName(tableName)
                 .indexName(indexName)
                  .searchQuery(SearchQuery
                              .newBuilder()
                               .query(QueryBuilders.term("col_date", "2012-12-12 12:10:03.123")).build())
                  .build());
}
```

Use the DateTimeFormatter method by installing Java Development Kit (JDK) V8 or later to verify the date format.

```
⑦ Note
Errors may exist if you use this method to verify the date format in some time zones.
import java.time.format.DateTimeFormatter;
public void testFormatByJdk8() {
    DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss.SSS").parse("2012-12-12 12:10:03.123");
}
```

5.5.5.4. ARRAY and NESTED field types

In addition to basic data types such as LONG, DOUBLE, BOOLEAN, KEYWORD, TEXT, and GEOPOINT, search indexes support the following special field types: ARRAY and NESTED.

ARRAY type

The ARRAY type can be combined with basic data types such as LONG, DOUBLE, BOOLEAN, KEYWORD, TEXT, and GEOPOINT. For example, the combination of LONG with ARRAY is used to specify arrays of the LONG INTEGER type. LONG ARRAY fields can contain multiple long integers. If a query matches a component of an array, the corresponding row is returned.

The following table describes the combination of ARRAY with basic data types

Combination	Description
LONG ARRAY	An array of long integers. Example: [1000, 4, 5555].
BOOLEAN ARRAY	An array of Boolean values. Example: [true, false].
DOUBLE ARRAY	An array of double-precision floating-point numbers. Example: [3.1415926, 0.99].
KEYWORD ARRAY	An array of strings in the JSON ARRAY format. Example: [\"Hangzhou\", \"Xi'an\"].
TEXT ARRAY	An array of text in the JSON ARRAY format. Example: [\"Hangzhou\", \"Xi'an\"]. In most cases, data of the TEXT type does not use the JSON ARRAY format.
GEOPOINT ARRAY	An array of coordinate pairs that consist of latitudes and longitudes. Example: [\"34.2, 43.0\", \"21.4, 45.2\"].

The ARRAY type is supported only in search indexes but not in data tables. If the data type of a field in a search index is a combination of ARRAY with a basic data type such as LONG or DOUBLE, the field in the data table for which the search index is created must be of the STRING type, and the field in the search index must be of the corresponding basic data type. For example, if the price field is of the DOUBLE ARRAY type, the price field in the data table for the search index must be of the STRING type. The price field in the search index must be of the STRING type. The price field in the search index must be of the STRING type. The price field in the search index must be of the STRING type. The price field in the search index must be of the DOUBLE type, and the isArray attribute must be set to true for the price field in the search index.

NESTED type

Data of the NESTED type is nested documents. Nested documents are used when a row of data (document) contains multiple child rows (child documents). Multiple child rows are stored in a NESTED field. You must specify the schema of child rows in the NESTED field. The schema must include the fields of the child rows and the attributes of each field. The NESTED type is similar to the ARRAY type. However, NESTED fields support more features.

NESTED fields are written as strings to data tables. NESTED fields in search indexes are of the JSON ARRAY type. Example: [{"tagName":"tag1", "score":0.8}, {"tagName":"tag2", "score":0.2}].

() Important

Even if a row contains a single child row, the written strings must be of the JSON ARRAY type.

• Create a single-level NESTED field

You can create a single-level NESTED field in the Tablestore console or by using Tablestore SDKs.

This section provides an example on how to create a single-level NESTED field by using Tablestore SDK for Java. In this example, a NESTED field named tags is used. Each child row contains two fields. The following figure shows the detailed information

Field Name	Field Type	Array	Index	Sort Statistics	Storage	Tokenization	Parameter
id	STRING	No	Yes	Yes	Yes		
tags	Nested Document	No	No	No	No		
tagName	STRING	No	Yes	Yes	Yes		
score	Floating Point	No	Yes	Yes	Yes		

• Field name: tagName. Type: KEYWORD.

Field name: score. Type: DOUBLE.

The following data samples are written to the data table: [{"tagName":"tag1", "score":0.8}, {"tagName":"tag2", "score":0.2}] .

// Construct the schema of the fields in the child rows.

List<FieldSchema> subFieldSchemas = new ArrayList<FieldSchema>();

subFieldSchemas.add(new FieldSchema("tagName", FieldType.KEYWORD)

- .setIndex(true).setEnableSortAndAgg(true)); subFieldSchemas.add(new FieldSchema("score", FieldType.DOUBLE)
- .setIndex(true).setEnableSortAndAgg(true));

// Specify that the schema of the fields in the child rows is used as the schema of the NESTED field. FieldSchema nestedFieldSchema = new FieldSchema("tags", FieldType.NESTED) .setSubFieldSchemas(subFieldSchemas);

· Create a multi-level NESTED field

You can create a multi-level NESTED field only by using Tablestore SDKs.

This section provides an example on how to create a multi-level NESTED field by using Tablestore SDK for Java. In this example, a NESTED field named user is used. Each child row contains three fields of basic data types and one NESTED field.

- Field name: name. Type: KEYWORD.
- Field name: age. Type: LONG.
- Field name: phone. Type: KEYWORD.
- Name of the NESTED field: address. Names of the fields contained in each child row: province, city, and street. All fields contained in each child row are of the KEYWORD type.

The following data samples are written to the table: [{"name":"Zhangsan", "age":20, "phone":"13900006666", "address":[{"province":"Zhejiang Province","city":"Hangzhou City","street":"No. 1201, Xingfu Community, Yangguang Avenue"}]}] .

// Construct the schema of the three fields in the child rows of the address NESTED field. The path specified by user.address can be used to query data of fields in a child row.

List<FieldSchema> addressSubFiledSchemas = new ArrayList<>();

addressSubFiledSchemas.add(new FieldSchema("province",FieldTvpe,KEYWORD));

addressSubFiledSchemas.add(new FieldSchema("city",FieldType.KEYWORD)); addressSubFiledSchemas.add(new FieldSchema("street",FieldType.KEYWORD));

// Construct the schema of the fields in the child rows of the user NESTED field. Each child row contains three fields of basic data types a nd one NESTED field named address. The path specified by user can be used to query data of fields in a child row.

- List<FieldSchema> subFieldSchemas = new ArrayList<>();
- subFieldSchemas.add(new FieldSchema("name",FieldType.KEYWORD))
- subFieldSchemas.add(new FieldSchema("age",FieldType.LONG)); subFieldSchemas.add(new FieldSchema("phone",FieldType.KEYWORD));

subFieldSchemas.add(new_FieldSchema("address",FieldTvpe.NESTED).setSubFieldSchemas(addressSubFiledSchemas));

// Specify that the schema of the fields in the child rows of the user NESTED field is used as the schema of the address NESTED field. List<FieldSchema> fieldSchemas = new ArrayList<>();

fieldSchemas.add(new FieldSchema("user",FieldType.NESTED).setSubFieldSchemas(subFieldSchemas));

The NESTED type has the following limits:

- Search indexes that contain NESTED fields do not support the IndexSort feature. The IndexSort feature can improve query performance in many scenarios.
- If you use a search index that contains NESTED fields to query data and pagination is required, you must specify a sorting method to return data that meets the query conditions. Otherwise, Tablestore does not return nextToken because only part of data that meets the query conditions is read.

• Nested queries provide lower performance than other types of queries.

Except for the preceding limits, you can perform all types of queries, sorting, and aggregations on data of the NESTED type.

5.5.5.5. Tokenization

After you specify a tokenization method for TEXT fields, Tablestore tokenizes field values into multiple tokens based on the tokenization method. You cannot specify tokenization methods for non-TEXT fields

You can use match queries and match phrase queries to query TEXT fields. In rare cases, you can also use term queries, terms queries, prefix queries, and wildcard queries.

Tokenization methods

The following tokenization methods are supported: single-word tokenization, delimiter tokenization, minimum semantic unit-based tokenization, and fuzzy tokenization.

Single-word tokenization

This tokenization method applies to all natural languages such as Chinese, English, and Japanese. The default tokenization method for TEXT fields is single-word tokenization.

After single-word tokenization is specified as the tokenization method, Tablestore performs tokenization based on the following rules:

- Chinese texts are tokenized based on each Chinese character. For example, "杭州" is tokenized into "杭" and "州". You can perform a match query or match phrase query and set the keyword to "杭" to query the data that contains "杭州".
- Letters or digits are tokenized based on spaces or punctuation marks. Uppercase letters are converted to lowercase letters. For example, "Hang Zhou" is tokenized into "hang" and "zhou". You can perform a match query or match phrase query and set the keyword to "hang", "HANG", or "Hang" to query the rows that contain "Hang Zhou".
- Alphanumeric characters such as model numbers are also separated by spaces or punctuation marks. By default, the letters and digits in
 alphanumeric characters are not separated. For example, "IPhone6" can only be tokenized into "IPhone6". If you perform a match query or match
 phrase query to query the rows that contain "IPhone6", you must set the keyword to "iphone6". No results are returned if you use "iphone".

The following table describes the parameters for single-word tokenization.

Parameter	Description
caseSensitive	Specifies whether to enable case sensitivity. Default value: false. If you set the parameter to false, all letters are converted to lowercase letters. To prevent Tablestore from converting letters to lowercase letters, you can set the parameter to true.
delimitWord	Specifies whether to tokenize alphanumeric characters. Default value: false. If you need to separate letters from digits, set the parameter to true. This way, "iphone6" is tokenized into "iphone" and "6".

Delimiter tokenization

Tablestore provides general dictionary-based tokenization. However, custom dictionaries are required for tokenization in particular industries. To meet this requirement, Tablestore provides delimiter tokenization. You can perform tokenization by using custom methods, use delimiter tokenization, and then write data to Tablestore.

Delimiter tokenization applies to all natural languages such as Chinese, English, and Japanese.

After delimiter tokenization is specified as the tokenization method, the system tokenizes field values based on the specified delimiter. For example, the value of a field is "badminton,ping pong,rap" in a row. The delimiter is set to a comma (,). The value is tokenized into "badminton", "ping pong", and "rap", and an index is created on the field. If you use a match query or match phrase query to query "badminton", "ping pong", "rap", or "badminton,ping pong", the row is returned.

The following table describes the parameters for delimiter tokenization.

Parameter	Description
delimiter	 The delimiter. The default delimiter is a whitespace character. You can specify a custom delimiter. If you create a search index for a data table, the delimiter specified for field tokenization must be the same as the delimiter that is used to write data to the data table. Otherwise, data may not be returned. If the custom delimiter is a special character such as a number sign (#) or a tilde (~), concatenate the delimiter with an escape character (\). Example:

• Minimum semantic unit-based tokenization

This tokenization method applies to the Chinese language in full-text search scenarios.

After minimum semantic unit-based tokenization is specified as the tokenization method, Tablestore tokenizes the values of TEXT fields into the minimum number of semantic units when Tablestore performs a query.

• Maximum semantic unit-based tokenization

This tokenization method applies to the Chinese language in full-text search scenarios.

After maximum semantic unit-based tokenization is specified as the tokenization method, Tablestore tokenizes the values of TEXT fields into the maximum number of semantic units when Tablestore performs a query. However, different semantic units may contain the same character. The total length of the tokenized words is longer than the length of the original text. Therefore, the data volume of the index is increased.

This tokenization method can generate more tokens and increase the probability that the rows meet the query conditions. However, the data volume of the index is increased. A match query is more suitable than a match phrase query if the tokenization method is maximum semantic unit-based tokenization. If you perform a match phrase query, data may not be returned due to overlapping tokens because the keyword is also tokenized by using the maximum semantic unit-based tokenization method.

Fuzzy tokenization

This tokenization method applies to all natural languages such as Chinese, English, and Japanese in scenarios that involve short text content, such as titles, movie names, book titles, file names, and directory names.

The combination of fuzzy tokenization and match phrase queries can be used to return query results at a low latency, which provides better query performance than wildcard queries. However, the data volume of the index is increased.

After fuzzy tokenization is specified as the tokenization method, Tablestore performs tokenization by using the N-gram counting component. The number of characters of a generated token falls within the range that is specified by the minChars and maxChars parameters. For example, the fuzzy tokenization method is used for drop-down suggestion.

Fuzzy tokenization converts the field values to lowercase letters. Therefore, fuzzy tokenization is case-insensitive and is similar to the LIKE operator in SQL statements.

To perform a fuzzy query, you must perform a match phrase query on the columns for which fuzzy tokenization is used.

- Limits
 - You can use fuzzy tokenization to tokenize the values of a TEXT field that is equal to or smaller than 1,024 characters in length. If a TEXT field exceeds 1,024 characters in length, excess characters are truncated and discarded. Only the first 1,024 characters are tokenized.
- To prevent excessive volume increase of index data, the difference between the values of the maxChars and minChars parameters must not
 exceed 6.
- Parameters

Parameter	Description
minChars	The minimum number of characters of a token. Default value: 1.
maxChars	The maximum number of characters of a token. Default value: 7.

Comparison

The following table compares the tokenization methods from multiple dimensions.

Item	Single-word tokenization	Delimiter tokenization	Minimum semantic unit-based tokenization	Maximum semantic unit-based tokenization	Fuzzy tokenization
Increase of the data volume	Small	Small	Small	Medium	Large
Relevance	Weak	Weak	Moderate	Relatively strong	Relatively strong
Applicable languages	All	All	Chinese	Chinese	All
Length limit	None	None	None	None	1,024 characters
Recall rate	High	Low	Low	Medium	Medium

5.5.6. Advanced features

5.5.6.1. Virtual columns

When you use the virtual column feature, you can modify the schema of a search index or create a search index to query data of new fields and new field types. You do not need to modify the storage schema and the data stored in Tablestore.

Objectives

The virtual column feature allows you to map a column in a data table to a virtual column in a search index when you create the search index. The type of the virtual column can be different from that of the column in the data table. This allows you to create a column without the need to modify the table schema and data. The new column can be used to accelerate queries or can be configured with different analyzers.

• Different analyzers supported for a field of the TEXT field

A single column of the STRING type can be mapped to multiple columns of the TEXT type of a search index. Different columns of the TEXT type use different tokens to meet various business requirements.

Query acceleration

You do not need to cleanse data or perform reindexing. You need to only map the required columns of a data table to the columns in a search index. The column types can be different between the data table and the search index. This improves the query performance in some scenarios. For example, you can convert the numeric type to the KEYWORD type to improve the performance of a term query. You can also convert the STRING type to the numeric type to improve the performance of a range query.

Usage notes

• The following table describes the data type conversion between virtual columns and columns in data tables.

Field type in data tables	Field type of virtual columns
STRING	KEYWORD and KEYWORD ARRAY
STRING	TEXT and TEXT ARRAY
STRING	LONG and LONG ARRAY
STRING	DOUBLE and DOUBLE ARRAY
STRING	GEOPOINT and GEOPOINT ARRAY
LONG	KEYWORD

LONG	ТЕХТ
DOUBLE	KEYWORD
DOUBLE	ТЕХТ

• Virtual columns can be used only in query statements and cannot be specified as the values of the ColumnsToGet parameter to return the values in the virtual columns, you can specify that the system returns the source columns of the virtual columns.

Use virtual columns in the Tablestore console

- After you specify a field as a virtual column when you create a search index in the Tablestore console, you can use the virtual column to query data. 1. Log on to the Tablestore console
- 2. On the **Overview** page, click the name of the instance that you want to manage or click **Manage Instance** in the **Actions** column of the instance.
- 3. In the **Tables** section of the **Instance Details** tab, click the name of the data table for which you want to create a search index. On the Manage
- Table page, click **Indexes**. You can also click **Indexes** in the **Actions** column of the data table.
- 4. On the Indexes tab, click Create Search Index.

5. In the Create Index dialog box, specify virtual columns when you create a search index.

- i. By default, the system generates a name for the search index. You can also specify a name for the search index.
- ii. Configure the Schema Generation Type parameter.
 - If you set the **Schema Generation Type** parameter to **Manual**, you must specify the field names, select field types, and specify whether to turn on Array.
 - If you set the Schema Generation Type parameter to Auto Generate, the system automatically uses the primary key columns and attribute
 columns of the data table as index fields. You can select field types and specify whether to turn on Array based on your business requirements.

? Note

The values of the **Field Name** and **Field Type** parameters must match those of the data table. For more information about the mappings between the field types in data tables and those in search indexes, see **Data type mappings**.

iii. Create a virtual column.

() Important

To create a virtual column, the data table must contain the source field and the data type of the source field must match that of the virtual column.

a. Click Add an Index Field.

- b. Configure the Field Name and Field Type parameters.
- c. Turn on Virtual Field. Configure the Index Field Name parameter.

iv. Click OK.

After the search index is created, find the search index in the Indexes section and click **Index Details** in the **Actions** column. In the Index Details dialog box, you can view the information about the search index, such as the metering information and index fields.

? Note

To modify the time to live (TTL) of the search index, click **Modify** next to the Time to Live parameter in the **Index Meters** section. For more information, see Lifecycle management.

6. Use the virtual column to query data.

- i. Click Manage Data in the Actions column of the virtual column.
- ii. In the Search dialog box, configure the required parameters
 - a. By default, the system returns all columns. To return specific attribute columns, turn off **All Columns** and specify the attribute columns that you want to return. Separate multiple attribute columns with commas (,).
 - b. Select index fields from the drop-down list. Click Add each time you select an index field. Configure the Query Type and Value parameters.
 - c. By default, the query results are sorted based on the primary key. To sort the query results based on index fields, turn on **Sort**, add the index fields based on which the query results are sorted, and specify a sorting method.

d. Click OK

Data that meets the filter conditions is displayed in the specified order on the **Indexes** tab.

Use Tablestore SDKs to manage virtual columns

After you specify a field as a virtual column when you use a Tablestore SDK to create a search index, you can use the virtual column to query data.

1. Create a search index and specify virtual columns

Parameters

For more information about the parameters, see the "Step 1: Create a search index" section of the Search indexes topic.

- Example
- The following sample code provides an example on how to create a search index that contains the Col_Keyword and Col_Long columns. In this example, a virtual column named Col_Keyword_Virtual_Long is created for the Col_Keyword column, and a virtual column named Col_Long_Virtual_Keyword is created for the Col_Keyword_Virtual_Long is mapped to the Col_Keyword column in the data table, and the Col_Long_Virtual_Keyword column is mapped to the Col_Long column in the data table.

	private static void createSearchIndex(SyncClient client) {
	CreateSearchIndexRequest request = new CreateSearchIndexRequest();
	request.setTableName(tableName); // Specify the name of the data table.
	request.setIndexName(indexName); // Specify the name of the search index.
	IndexSchema indexSchema = new IndexSchema();
	indexSchema.setFieldSchemas(Arrays.asList(
	new FieldSchema("Col Keyword", FieldType.KEYWORD) // Specify the name and type of the Col Keyword column.
	new FieldSchema("Col Keyword Virtual Long", FieldType.LONG) // Specify the name and type of the Col Keyword Virtual Long column.
	.setVirtualField(true) // Specify whether the Col Keyword Virtual Long column is a virtual column.
	.setSourceFieldName("Col Keyword"), // Specify the name of the source field to which the virtual column is mapped in the data t
	able.
	new FieldSchema("Col Long", FieldType.LONG),
	new FieldSchema("Col_Long_Virtual_Keyword", FieldType.KEYWORD)
	.setVirtualField(true)
	.setSourceFieldName("Col_Long")));
	request.setIndexSchema(indexSchema);
	client.createSearchIndex(request); // Call the client to create the search index.
	}
2. 1	Jse the virtual columns to query data.
(Query the rows in which the value of the Col_Long_Virtual_Keyword column matches 1000 from the data table for which the search index is created.
	Specify that the rows that meet the query condition and the total number of rows that meet the query condition are returned.
	nrivate static void guery(SuncClient client) /
	SearchOnery searchOnery = new SearchOnery().
	Searching terms (Dierv = new TermsChierv(): // Set the gierv parameter to TermsChierv.
	termsQuery.setFieldName("Col Long Virtual Keyword"): // Specify the name of the field that you want to guery.
	termsQuery.addTerm(ColumnValue.fromString("1000")): // Specify the value that you want to match.
	searchOuerv.setOuerv(termsOuerv):
	searchOuerv.setGetTotalCount(true): // Specify that the total number of rows that meet the guery condition is returned.
	SearchRequest searchRequest = new SearchRequest("tableName", "indexName", searchOuerv);
	SearchRequest.ColumnsToGet columnsToGet = new SearchRequest.ColumnsToGet();
	columnsToGet.setReturnAll(true); // Set the ReturnAll parameter to true to return all columns excluding the virtual columns.
	searchRequest.setColumnsToGet(columnsToGet);
	SearchResponse resp = client.search(searchRequest);
	System.out.println("TotalCount: " + resp.getTotalCount()); // Display the total number of rows that meet the guery condition instead of
	the number of returned rows.
	System.out.println("Row: " + resp.getRows());
)

5.5.6.2. Dynamically modify schemas

You can dynamically modify the schema of a search index. For example, you can add, update, or delete index columns for the search index, and modify the routing keys of the search index.

Feature description

Data tables of Tablestore are schema-free. However, search indexes have rigid schemas. When you create a search index, you must specify the columns you want to add to the search index. Then, you can query these columns when you use the search index to query data. To adapt to business changes and optimize performance, an increasing number of users need to modify the schemas of search indexes. Tablestore allows you to dynamically modify the schemas of search indexes in the following scenarios:

- · Add index columns: You can add index columns if your business requires more columns for queries.
- Update index columns: Modify the analyzer of a TEXT field.
- Delete index columns: You may need to remove unnecessary columns added when you create a search index.
- Modify routing keys: You can specify routing keys to reduce read workloads and improve query efficiency.

The following process describes how to dynamically modify a schema with ease. The entire process does not affect business. You do not need to change business code.

- 1. Add, modify, or remove the index columns of a search index to create a canary index.
- 2. Wait until the existing and incremental data of the data table is synchronized to the canary index and the synchronization progress is the same as the synchronization progress of the search index.
- 3. Use A/B testing to gradually switch traffic to the canary index and wait until all traffic is switched to the canary index.
- 4. After you verify that the canary index works normally, switch the schemas between the source index and the canary index.

5. Delete the source index schema.

Procedure

1. Go to the Indexes tab.

- i. Log on to the Tablestore console.
- ii. On the **Overview** page, click the name of the instance that you want to manage, or click **Manage Instance** in the **Actions** column of the instance that you want to manage.
- iii. In the Tables section of the Instance Details tab, click the name of the data table and then click the Indexes tab. You can also click Indexes in the Actions column that corresponds to the data table.
- 2. Create a canary index based on the source index.
- i. On the Indexes tab, click Change Schema in the Actions column that corresponds to the search index.
- ii. In the **Reindex** dialog box, add, modify, or delete index fields. Move the pointer over **Reindexing Procedure** to view the process to change the schema of a search index.
- iii. Click OK.
- iv. In the Index Comparison message, compare the schema information between the source index and the canary index. After you confirm the information, click OK.
- 3. View the index synchronization information.
- The existing data synchronization and incremental data synchronization stages are required for the canary index. Before data is synchronized, the system displays **Yes**, but the operation may cause security risks. In this case, you cannot perform the switchover. When the synchronization

progress of the canary index is the same as that of the source index, the system displays Yes. The operation is secure. You can perform subsequent operations.

- i. Click the icon in front of the source index or click the name of the source index
- The system displays the canary index of the source index.
- ii. Click Use Gray Index in the Actions column that corresponds to the canary index.
- iii. In the Use Gray Index dialog box, view the synchronization information of the indexes.
- 4. After data is synchronized for the indexes, set weights to perform A/B testing. A/B testing allows you to allocate traffic to the source index and the canary index based on proportions and verify the effects of changes to the
- schema. You can perform subsequent operations only when all traffic is switched to the canary index. i. In the Operations section of the Use Gray Index dialog box, adjust the slider to control the weights for the source index and the canary index. Click Set Weight.
- ii. In the Set Weight message, view the weight data and the schema comparison information.
- iii. After you confirm the information, click Set Weight.
- iv. In the Set Weight message, click OK
- 5. After all traffic for queries is switched to the canary index, switch the schemas between the source index and the canary index. After you switch the schemas, the name of the source index is associated with the new schema. The name of the canary index is associated with the source index schema. All traffic is switched to query the source index whose name is associated with the new schema.
 - i. In the Operations section of the Use Gray Index dialog box, click Switch Index.
- ii. In the Switch Index dialog box, check the schema information of the source index and the canary index. Click Confirm Switch.
- 6. You can delete the source index schema after you verify that new schema is correct. To delete the source index schema, we recommend that you wait for a period of time such as one day. In the **Use Gray Index** dialog box, click **Delete Source Search Index** to delete the source index schema.

Security

To prevent incorrect operations, Tablestore provides the rollback mechanism and switchover notes to minimize the risks caused by modifying schemas. Rollback mechanism

- When you dynamically modify the schema of a search index, you can roll back the modification.
- After you create a canary index, you can delete the canary index and create a new one if the schema of the canary index does not meet your expectations.
- When you perform A/B testing, you can configure weights to gradually switch traffic to the canary index. In this process, you can reset the weights anytime to switch traffic back to the source index if issues occur.
- After you switch the schemas between the source index and the canary index, you can cancel the switchover anytime to switch back the schemas if issues occur. Index switchover is the reverse of switchover cancellation.
- Switchover notes

If you switch traffic to a canary index when the synchronization progress of the canary index is slower than that of the source index, the data you query may not be the latest. In this case, Tablestore determines whether switchover can be performed based on the synchronization status and the last synchronization time of the source index and the canary index.

- If the following situations exist. Tablestore determines that switchover can be performed:
- The source index is in the full data synchronization stage. The canary index is in the full or incremental data synchronization stage. The synchronization progress of the canary index is the same as that of the source index.
- The source index and the canary index are in the incremental data synchronization stage. The last synchronization time of the source index is at
 most 60 seconds earlier than that of the canary index.

5.5.6.3. Fuzzy query

If you perform a wildcard query to search for the *word* string, you can use fuzzy tokenization and a match phrase query to ensure better query performance

Background information

Fuzzy queries are commonly performed in database business scenarios. For example, you can perform a fuzzy query to query file names and mobile numbers. To perform fuzzy queries in Tablestore, you can use the wildcard query feature of search indexes. The wildcard query feature is similar to the LIKE operator in MySQL. However, the string that is used for a wildcard query can contain only up to 20 characters, and the query performance decreases as the volume of data increases.

To resolve these issues, search indexes support fuzzy tokenization to ensure high-performance fuzzy queries. When you use the fuzzy tokenization feature, Tablestore does not limit the length of the string that is used for a query. However, if the field value exceeds 1,024 characters in length, the system truncates the field value and performs tokenization only for the first 1,024 characters.

Scenarios

You can select a method to perform a fuzzy query based on your business requirements.

• If you use *word* for a wildcard query, you can use fuzzy tokenization to perform a fuzzy query. For example, if you use "123" to query mobile numbers that contain 123 at any position, you can use fuzzy tokenization to perform a fuzzy query.

In this case, the fuzzy tokenization feature improves the query performance by more than 10 times than the wildcard query.

For example, a data table contains a column named file_name. The column is of the TEXT type and the tokenization method is fuzzy tokenization. If you use a search index to query the rows in which the value of the file_name column is 2021 woRK@Hangzhou , you must perform a match phrase guery and specify the tokens as consecutive substrings for the guery.

- The rows in which the value of the file name column is 2021 work@Hangzhou are returned if one of the following tokens is used for the query: 2021 , 20 , 21 , work , WORK , ${\tt @}$, Hang , zhou , Hangzhou , ${\tt and}$ @Hangzhou
- The rows in which the value of the file_name column is 2021 work@Hangzhou are not returned if one of the following tokens is used for the query: 21work , 2021Hangzhou , 2120 , and #Hangzhou
- · For other complex gueries, you can perform wildcard gueries

Use fuzzy tokenization for a fuzzy query

To use fuzzy tokenization for a fuzzy query, perform the following steps:

- Create a search index. When you create the search index, set the types of the fields on which the search index is created to TEXT and set the tokenization method to fuzzy tokenization (Fuzzy Analyzer). You can use the default values of other parameters
- 2. Perform a match phrase query by using the search index.

Example

The following sample code provides an example on how to use fuzzy tokenization to perform a fuzzy query:

package com.aliyun.tablestore.search.test; import com.alicloud.openservices.tablestore.SyncClient; import com.alicloud.openservices.tablestore.model.*; import com.alicloud.openservices.tablestore.model.search.*; import com.alicloud.openservices.tablestore.model.search.query.QueryBuilders; import org.junit.Test; import java.util.Arrays; import java.util.Collections; import static org.junit.Assert.assertEquals; public class Test { private static final Conf conf = Conf.newInstance("src/test/resources/conf.json"); private static final SyncClient ots = new SyncClient(conf.getEndpoint(), conf.getAccessId(), conf.getAccessKey(), conf.getInstanceName()); private static final String tableName = "analysis test"; private static final String indexName = "analysis_test_index"; GTest public void testFuzzyMatchPhrase() { // Delete the existing data table and indexes. TableStoreHelper.deleteTableAndIndex(ots, tableName); // Create a data table. TableStoreHelper.createTable(ots, tableName); // Define the schema of the data table. IndexSchema indexSchema = new IndexSchema(); indexSchema.setFieldSchemas(Collections.singletonList($\prime\prime$ Note: If you change the type of the name field for the query from KEYWORD to TEXT and specify the tokenization method for th e field, exceptions may occur during the query. // If you need to only query the rows in which the value of the name field matches the *abc* string, set the type of the name field to TEXT. The KEYWORD type is not required. new FieldSchema("name", FieldType.TEXT).setAnalyzer(FieldSchema.Analyzer.Fuzzy))); // Create a search index TableStoreHelper.createIndex(ots, tableName, indexName, indexSchema); // Write a row of data to the data table. PrimaryKey primaryKey = PrimaryKeyBuilder.createPrimaryKeyBuilder() .addPrimaryKeyColumn("pk1", PrimaryKeyValue.fromString("1")) .addPrimaryKeyColumn("pk2", PrimaryKeyValue.fromLong(1)) .addPrimaryKeyColumn("pk3", PrimaryKeyValue.fromBinary(new byte[]{1, 2, 3})) .build(); RowPutChange rowPutChange = new RowPutChange(tableName, primaryKey); // Add an attribute column to the data table. rowPutChange.addColumn("name", ColumnValue.fromString("TheBlindMelody1024x768P.mp4")); PutRowRequest request = new PutRowRequest(rowPutChange); ots.putRow(request); $\ensuremath{{\prime}}\xspace$ // Wait until the row of data is synchronized to the search index. TableStoreHelper.waitDataSync(ots, tableName, indexName, 1); // Use *abc* for the fuzzy query. assertMatchPhraseQuery(ots, tableName, indexName, "name", "The", 1); assertMatchPhraseQuery(ots, tableName, indexName, "name", "TheBlind", 1); assertMatchPhraseQuery(ots, tableName, indexName, "name", "TheBlind", 0); assertMatchPhraseQuery(ots, tableName, indexName, "name", "TheBlindMelody102", 1); assertMatchPhraseQuery(ots, tableName, indexName, "name", "TheBlindMelody1024", 1); assertMatchPhraseQuery(ots, tableName, indexName, "name", "TheBlindMelody1024x", 1); assertMatchPhraseQuery(ots, tableName, indexName, "name", "TheBlindMelody1024x7", 1); assertMatchPhraseQuery(ots, tableName, indexName, "name", "TheBlindMelody1024x768P.mp4", 1); assertMatchPhraseQuery(ots, tableName, indexName, "name", "24x768P.mp4", 1); assertMatchPhraseQuery(ots, tableName, indexName, "name", "24x76 8P.mp4", 0); assertMatchPhraseQuery(ots, tableName, indexName, "name", "24x7 P.mp4", 0); // Perform a match phrase guery. public static void assertMatchPhraseQuery(SyncClient ots, String tableName, String indexName, String fieldName, String searchContent, long exceptCount) { SearchRequest searchRequest = new SearchRequest(); searchRequest.setTableName(tableName); searchRequest.setIndexName(indexName); SearchQuery searchQuery = new SearchQuery(); // Perform a match phrase query to query data that matches the tokens. searchQuery.setQuery(QueryBuilders.matchPhrase(fieldName, searchContent).build()); searchQuery.setLimit(0); // Specify that the total number of rows that meet the query conditions is returned. If you are not concerned about the total number of rows that meet the query conditions, set this parameter to false for better query performance searchQuery.setGetTotalCount(true); searchRequest.setSearchQuery(searchQuery); SearchResponse response = ots.search(searchRequest); assertEquals(String.format("field:[%s], searchContent:[%s]", fieldName, searchContent), exceptCount, response.getTotalCount()); } }

6.Hybrid Disaster Recovery 6.1. What is Hybrid Disaster Recovery?

Hybrid Disaster Recovery (HDR) is a service that integrates on-premises backup and disaster recovery in the cloud. It serves the needs of enterpriselevel applications in data centers. HDR provides disaster recovery services with a recovery point objective (RPO) of a few seconds and a recovery time objective (RTO) of a few minutes for enterprise-level applications that are deployed in data centers and on Alibaba Cloud. HDR can effectively ensure data security and business continuity. This service is applicable to the following scenarios:

- Scenario 1: HDR provides cross-zone disaster recovery capabilities to meet different business requirements. If the primary system is faulty, your
 business system is switched to the disaster recovery system. This effectively prevents system failures caused by regional disasters, ensures business
 availability, and meets the RPO and RTO goals of your business. In this scenario, HDR supports the following technologies:
- Continuous data replication (CDR)

You must install an agent on each protected instance. This technology has requirements for the operating systems of protected instances. You must check whether the operating systems of your instances are supported.

Async replication

Async replication is implemented on disks without the need to install an agent on each protected instance.

Scenario 2: HDR provides cross-cloud disaster recovery capabilities by deploying the primary and secondary clouds in different regions. If the
primary cloud is faulty, your business system is switched to the secondary cloud. This effectively prevents system failures caused by exceptions on
cloud instances, ensures business availability, and meets the RPO and RTO goals of your business. In this scenario, HDR supports async replication.

6.2. Getting started

6.2.1. Procedure

This topic describes the basic procedure of disaster recovery in Hybrid Disaster Recovery (HDR).

To implement disaster recovery for key applications, perform the following steps:

1. Step 1: Plan resources

Before you implement disaster recovery, you must determine the region and zone in which you want to create a disaster recovery site. Create a virtual private cloud (VPC) and vSwitches at the disaster recovery site.

2. Step 2: Create a site pair

Configure the CIDR blocks to be used at the disaster recovery site. During the test, you can use the default configurations to create a VPC and vSwitches at the disaster recovery site. You can also configure the same VPC CIDR block and vSwitch CIDR block for the production site and disaster recovery site. During actual disaster recovery, you can configure CIDR blocks as required.

3. Step 3: Configure network and security settings

Map resources, including vSwitches and security groups.

4. Step 4: Create a protection group

If you use async replication as the replication technology, you must create a protection group.

- 5. Step 5: Add instances to be protected
- Add instances to be protected to the protection group.
- 6. Step 6: Start replication
- Start disaster recovery protection. Data is replicated from the production site to the disaster recovery site.
- 7. Step 7: Perform a failover
- Switch After Data Synchronization

During the failover, HDR stops the protected instances in the protection group, and performs the final data synchronization after all the protected instances are stopped. The failover starts after the data is synchronized. This ensures that the data at the disaster recovery site is the same as that at the production site. This type of failover applies to scenarios such as planned disaster recovery drills and business migration.

Switch Now

During the failover, HDR attempts to stop the protected instances in the protection group. HDR does not wait until all the protected instances are stopped or perform the final data synchronization. Some data may be lost within the recovery point objective (RPO) range. This type of failover applies to scenarios in which a fault cannot be rectified within a short period of time at the production site and business must be immediately switched to the disaster recovery site.

6.2.2. Log on to the HDR console

This topic describes how to log on to the Hybrid Disaster Recovery (HDR) console.

Prerequisites

- The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel.
- A web browser is available. We recommend that you use Google Chrome.
- 1. Enter the URL of the Apsara Uni-manager Management Console in the address bar and press the Enter key.
 - Note
 You can click the current language in the upper-right corner to switch to another language.
- 2. Enter your username and password.

Obtain the username and password from an operations administrator.

?	Note

• First logon

The first time that you log on to the Apsara Uni-manager Management Console, you need to change the password of your account. The password must be 10 to 32 characters in length. It must contain all of the following character types: uppercase letters, lowercase letters, digits, and special characters. Supported special characters include: ! @ # \$ %

• Forget password

If you have forgotten your password, click Forgot Password. On the page that appears, enter the username of your account, the email address that was used to create the account, and the CAPTCHA code. Then, the system sends a link for resetting the password to the specified email address.

- 3. Click Log On.
- 4. If multi-factor authentication (MFA) is enabled for your account, perform the corresponding operations in the following scenarios:
 - You log on to the Apsara Uni-manager Management Console for the first time after MFA is enabled by the administrator.
 - a. On the Bind Virtual MFA Device page, bind an MFA device.
 - b. Enter the username and password again as in Step 2 and click $\ensuremath{\text{Log On}}$.
 - c. Enter a six-digit MFA verification code and click Authenticate.
 - You have enabled MFA and bound an MFA device.

Enter a six-digit MFA verification code and click Authenticate.

⑦ Note

For information about how to bind and enable MFA, see Manage MFA in Apsara Uni-manager Management Console User Guide .

5. In the top navigation bar, choose **Products > Storage > Hybrid Disaster Recovery**.

6. Select an organization and a region, and then click Access as Administrator to go to the HDR console.

6.2.3. Terms

This topic describes the terms that are used in Hybrid Disaster Recovery (HDR).

Term	Description
site pair	A pair of sites that reside in different regions or zones. A site pair has one or more protection groups. Disaster recovery is implemented only in the forward direction for the protection groups in a site pair. For example, disaster recovery is performed from Protection Group A to Protection Group B, and the forward protection is initiated from Region 1 to Region 2. Disaster recovery is performed from Protection Group D, and the forward protection Group D, and the forward protection is initiated from Region 2 to Region 1. In this case, you must create two site pairs. A protection group can belong to only one site pair. Only one replication technology can be used for one site pair.
protection group	 A group of protected Elastic Compute Service (ECS) instances. You can use one plan to perform operations on multiple ECS instances in a protection group at the same time. You can create a regular group or a consistent group. In a regular protection group, ECS instances do not have associations with each other. Only one underlying technology can be applied to the ECS instances in a protection group to implement disaster recovery: CDR or async replication. You must determine the underlying technology when you create a protection group. The normal states of a protection group include Enabling Replication, Replicating Full Data, Replicating Incremental Data, Failover in Progress, Failover Completed, Reverse Replicating, Failback in Progress, and Failback Completed. The abnormal states include Replication Error, Failover Failed, and Failback Failed. A failover is performed for all the protected ECS instances in a protection group. Therefore, the roles of all the protected ECS instances in a protection group.
protected instance	An ECS instance or a database that is protected. Database protection will be supported in the future. Roles include primary and secondary. Primary instances are instances on which services are running, and secondary instances are instances that are used for disaster recovery.
production site	The zone or region in which your production business initially operates.
disaster recovery site	The zone or region for disaster recovery of your production business.
failover	A disaster recovery process that resumes your business at the disaster recovery site if your application fails.
failback	A disaster recovery process that switches your business from the disaster recovery site to the production site after the production site is recovered.
recovery point objective (RPO)	The maximum tolerable data loss during a disaster. For example, if an RPO is 15 minutes, data that is generated within the most recent 15 minutes cannot be recovered on the cloud.
recovery time objective (RTO)	The amount of time required to recover an application from a disaster on the cloud.
forward protection	The replication direction of protection groups and ECS instances. In forward protection, data and services are replicated from the production site to the disaster recovery site.
reverse protection	The replication direction of protection groups and ECS instances. After a failover, the disaster recovery site (Site B) becomes the new production site, and the production site (Site A) becomes the new disaster recovery site. In this case, after the replication is started, data is replicated from Site B to Site A. The reverse protection takes effect on the site pair. After a failback, Site A becomes the production site a disaster recovery site again. In this case, after the replication is started, data is replicated from Site B becomes the from Site B to Site A to Site A to Site B. The forward protection resumes on the site pair.

6.3. Disaster recovery in the cloud

6.3.1. Cross-zone disaster recovery

6.3.1.1. Process overview

When a production site encounters force majeure events such as a fire disaster or an earthquake or equipment failures such as software or hardware failures, applications may fail to run in a short period of time. In this case, Hybrid Disaster Recovery (HDR) provides cross-zone disaster recovery for you to back up application data and run applications in another zone to deal with failures in a single zone at the required recovery point objective (RPO) and recovery time objective (RTO)

Solution overview

Cross-zone disaster recovery supports the following two replication technologies:

Continuous Data Replication (CDR)

If you use this replication technology, you must install an agent on the protected instance. This replication technology has requirements for the operating system of the protected instances. You must check whether the operating systems of your instances are supported.

Async replication

Async replication is implemented on disks without the need to install an agent on each protected instance.

The following table describes the differences between the two replication technologies.

? Note

The specifications and performance metrics described in the table are for reference only. The specifications and performance metrics vary based on the number of servers that are actually deployed, server performance, and network topology. We recommend that you contact Apsara Stack technical support to plan resources for disaster recovery.

Item	CDR	Async replication
Applicable scenarios	Disaster recovery for a single VM . If you do not mind intrusions into the system, you can use this replication technology.	Disaster recovery that ensures the consistency of VM groups . If you do not expect intrusions into the system, you can use this replication technology.
Intrusive to the system	Yes.	No.
Replication implementation	Installs an agent on the protected instance to embed an operating system and replicate the data written on the disk to a gateway in real time. The gateway stores the data in an Object Storage Service (OSS) bucket and then writes the data to the disk at the disaster recovery site.	Uses the async replication and snapshot mechanisms of Elastic Block Storage (EBS).
Recovery implementation	Supports multiple recovery points. A shadow Elastic Compute Service (ECS) instance and a gateway server are created for the protected ECS instance at the disaster recovery site. HDR reads data from the OSS bucket, writes the data to the ECS instance at the disaster recovery site, and then creates a recovery point based on the snapshot mechanism.	Supports only a single recovery point. HDR creates a recovery point by replicating the snapshot to the disaster recovery site.
Multiple recovery points	Supported. Recovery points can be created on an hourly basis.	Not supported. Only the last recovery point is supported.
Minimum RPO	Seconds to minutes.	2 minutes.
RTO	Minutes.	Minutes.
Single disk capacity	 Consistent with the ECS instance. If the system disk size does not exceed 500 GB, you can configure a whitelist. This way, the system disk size can be up to 2 TB. If the system disk size does not exceed 32 TB, the single disk capacity depends on the disk type of the ECS instance. For more information, see Developer Guide-Elastic Compute Service-API reference-CreateDisk. 	 Consistent with the ECS instance. If the system disk size does not exceed 500 GB, you can configure a whitelist. This way, the system disk size can be up to 2 TB. If the system disk size does not exceed 32 TB, the single disk capacity depends on the disk type of the ECS instance. For more information, see the "CreateDisk" topic in the API reference of the ECS developer guide.
Single disk write speed	Linux: • 4 KB: less than or equal to 10 MB/s. • 64 KB: less than or equal to 30 MB/s.	Less than or equal to 80 MB/s.
Protection group specifications	Each site pair supports one protection group by default. The total throughput of a protection group does not exceed 50 MB/s.	 The maximum number of disks supported by a replication pair-consistent group is 512. The maximum disk capacity supported by a replication pair-consistent group is 128 TB. Each site pair supports 512 protection groups. The total throughput of a protection group does not exceed the value of the number of nodes * 80 MB/s.
Maximum number of site pairs	1,000 within an account.	1,000 within an account.
Operating system	Limited. For more information, see Limits.	Unlimited. However, you must take into account the replacement of the system disks on ECS instances.

Limits on single VM specifications	Up to 16 disks (a system disk and 15 data disks).	Consistent with the limits of ECS instance type.
Cross-VM consistent group feature	Not supported.	Supported.
Disk type	Unlimited.	Premium performance disk.
Shared disk/encrypted disk	Not supported.	Not supported.
Data consistency	Crash consistency.	Crash consistency.
Sandbox disaster recovery drill	Supported.	Not supported.
Planned switchover with no data loss	Not supported.	Supported.
Common features of disaster recovery such as forward protection, reverse protection, failover, and failback	Supported.	Supported.
Zone-disaster recovery	Supported.	Supported.
Cross-region disaster recovery	Not supported.	Not supported.
Non-Alibaba Cloud to Alibaba Cloud	Not supported.	Not supported.
Direct dependency	ECS, OSS, and EBS.	ECS and EBS.
Disk configuration change	You can increase the number of disks or expand the capacity of a disk when the protection group is in the forward replication state. You cannot reduce the number of disks or shrink a disk.	Not supported. You cannot increase or reduce the number of disks, or attach or detach a disk. You cannot change the capacity of a disk.

Cross-cloud disaster recovery supports only the async replication technology.

Procedure

To perform disaster recovery on key applications, perform the following steps:

- Async replication
- i. Step 1: Plan resources

Before you perform disaster recovery, you must determine the region and zone in which you want to create a disaster recovery site. Create a virtual private cloud (VPC) and vSwitches at the disaster recovery site.

ii. Step 2: Create a site pair

Configure the CIDR blocks to be used at the disaster recovery site. During the test, you can use the default configurations to create a VPC and vSwitches at the disaster recovery site. You can also configure the same VPC CIDR block and vSwitch CIDR block for the production site and the disaster recovery site. During actual disaster recovery, you can configure CIDR blocks based on your business requirements.

- iii. Step 3: Configure network and security settings
- Map resources, including vSwitches and security groups.
- iv. Step 4: Create a protection group
 - To use the async replication technology, you must create a protection group.
- v. Step 5: Add instances to be protected
 - Add instances to be protected to the protection group.
- vi. Step 6: Enable replication
 - Start disaster recovery protection. Data is replicated from the production site to the disaster recovery site.
- vii. Step 7: Perform a failover
 - Switch After Data Synchronization

During a failover, HDR stops the protected instances in the protection group, and performs the final data synchronization after all the protected instances are stopped. The failover starts after the data is synchronized. This ensures that the data at the disaster recovery site is the same as that at the production site. This type of failover applies to scenarios such as planned disaster recovery drills and business migration.

Switch Now

During the failover, HDR attempts to stop the protected instances in the protection group. HDR does not wait until all the protected instances are stopped or perform the final data synchronization. Some data may be lost within the RPO range. This type of failover applies to scenarios in which a fault cannot be rectified within a short period of time at the production site and business must be immediately switched to the disaster recovery site.

- CDR
- i. Step 1: Plan resources

Before you perform disaster recovery, you must determine the region and zone in which you want to create a disaster recovery site. Create a VPC and vSwitches at the disaster recovery site.

ii. Step 2: Create a site pair

Configure the CIDR blocks to be used at the disaster recovery site. During the test, you can use the default configurations to create a VPC and vSwitches at the disaster recovery site. You can also configure the same VPC CIDR block and vSwitch CIDR block for the production site and the disaster recovery site. During actual disaster recovery, you can configure CIDR blocks based on your business requirements.

- iii. Step 3: Configure network and security settings
- Map resources, including vSwitches and security groups.
- iv. Step 5: Add instances to be protected Add instances to be protected to the protection group.
- v. Step 6: Enable replication
 - Start disaster recovery protection. Data is replicated from the production site to the disaster recovery site.
- vi. Step 7: Perform a failover

When a protected instance is in the Replicating state, you can perform operations such as disaster recovery drills, failover, reverse replication, and failback.

6.3.1.2. Async replication

6.3.1.2.1. Step 1: Plan resources

Before you perform disaster recovery, you must plan the computing, storage and network resources that are required for disaster recovery.

Computing resources

When you create Elastic Compute Service (ECS) instances, the disks must meet the following requirements:

- 1. Only premium performance disks are supported.
- 2. Disks are not encrypted.
- 3. Multi-attach is not supported.

Storage resources

During disaster recovery, all data at the production site is replicated to the disaster recovery site. Therefore, you must ensure that the disaster recovery site has sufficient storage resources. If you need to use a storage set to isolate resources for disaster recovery, configure the storage set at the disaster recovery site in advance.

Network resources

Create the virtual private clouds (VPCs), vSwitches, and security group rules that are required by the disaster recovery site.

6.3.1.2.2. Step 2: Create a site pair

Before you perform disaster recovery, you must create a site pair for centralized management of site resources. This topic describes how to create a site pair.

Prerequisites

vSwitches are created at the disaster recovery site. The disaster recovery site supports vSwitches that reside in the same virtual private cloud (VPC) as the primary site or a different VPC from the primary site.

Limits

Hybrid Disaster Recovery (HDR) imposes limits on Elastic Compute Service (ECS) instances that require disaster recovery. For more information, see Limits.

Procedure

- 1. Log on to the HDR console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click Site Pairs.
- 3. On the Site Pairs page, click Create Site Pair.
- 4. In the Create Site Pair panel, configure the parameters that are described in the following table, and click OK.

Category	Parameter	Description
Disaster recovery type	Туре	 The disaster recovery type. In this scenario, select Cross-zone Disaster Recovery. Cross-zone Disaster Recovery: replicates data from one zone to another zone in the specified region in real time. Cloud to Cloud: replicates cloud data from the production site to the disaster recovery site in real time.

Replication technology	Replication Technology	 The type of technology used to replicate data for disaster recovery. In this scenario, select Async Replication. Async Replication The async replication technology is implemented at the virtualization layer without the need to install an agent on the protected instance. This technology applies to consistent disaster recovery for a group of ECS instances. The applicable customers are those who can accept a recovery point objective (RPO) of a few minutes and do not expect intrusions into the system. CDR To use continuous data replication (CDR), you must install an agent on each protected instance. This replication technology is suitable for the scenario in which you want to implement disaster recovery for a single Elastic Compute Service (ECS) instance and can install software on the ECS instance. This technology has requirements for the operating systems of protected instances. You must check whether the operating systems of your instances are supported.
Resource set	The Organization's resource set	The resource set to which the local cloud site belongs. HDR automatically uses the resource sets specified in the organization mapping to manage computing, network, and storage resources based on the cloud site configurations of the production site and disaster recovery site.
Information of the production site and	Name	 The name of the production site or disaster recovery site. The name cannot exceed 60 characters in length. A recognizable name is required. Operations such as the deployment of a disaster recovery gateway must be performed after the site pair is created. The primary site is used to specify the location of the instances that require disaster recovery on the cloud. The computing and storage resources that are used by the disaster recovery site are created in the specified VPC.
disaster recovery site	Region	The region in which the production site or disaster recovery site resides. Disaster recovery data is restored in the region.
	VPC	The VPC that is used by the production site or disaster recovery site.
	Zone	The zone in which the production site or disaster recovery site resides.

After the site pair is created, the site pair is displayed on the **Site Pairs** page.

If the site pair is incorrectly configured, you can click Delete to the right of the site pair to delete the site pair and then create a site pair again.

6.3.1.2.3. Step 3: Configure network and security settings

This topic describes the infrastructure configurations for disaster recovery, including vSwitch mappings and security group mappings.

Create a vSwitch mapping

The type of the current site pair is **Cross-zone Disaster Recovery**, so you do not need to create a zone mapping. To create a vSwitch mapping, perform the following steps:

- 1. Log on to the Hybrid Disaster Recovery (HDR) console.
- For more information, see Log on to the HDR console. 2. In the left-side navigation pane, click **Site Pairs**.
- 2. In the left-side havigation pane, click **Site Pairs**.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Network & Security tab.
- 4. In the left sidebar, click vSwitch Mapping, and click Add vSwitch Mapping.
- 5. In the Add vSwitch Mapping dialog box, configure the parameters that are described in the following table, and click OK.

() Important

If the CIDR block of the vSwitch at the production site is different from that at the disaster recovery site, IP address inconsistency occurs when you perform a failover or failback.

Parameter	Description
Network of Production Site	The CIDR block of the vSwitch at the production site. Tor example, if the CIDR block of the vSwitch at the production site is 192.168.0.0/24 and the CIDR block of the vSwitch at the disaster recovery site is 192.168.1.0/24, IP addresses are mapped based on the following rule: 192.168.0.1 mapped to 192.168.1.1.
Network of DR Site	The CIDR block of the vSwitch at the disaster recovery site.

After the vSwitch mapping is created, it is displayed on the vSwitch Mapping tab.

Overview	Prote	ction Group Recovery Plan	Network & Security	Tasks						Edit Delete Site Pair
Zone Map…		Adr. EbsCadSitePair/InfrastructureTabitem.NetworkMapping.TipInfo								
vSwitch M… Security Gr…		Zone Mappino1 cn-wu		02-ь						Add vSwitch Mapping
		Network of Production Site		Network of DR Site		Status		Status Description		
		doctest-switch-a vsw-g	4	doctest-switch-b	/24	✓ Normal		The mapping status is normal.	Edit Delete	

If you want to edit or delete the created vSwitch mapping, click Edit or Delete on the right side of the vSwitch mapping.

Create a security group mapping

- 1. Log on to the HDR console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click Site Pairs.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Network & Security tab.
- 4. In the left sidebar, click Security Group Mapping, and click Add Security Group Mapping.
- 5. In the Add Security Group Mapping dialog box, configure the parameters that are described in the following table, and click OK.

Parameter	Description
Production Site Security Group	The security group that is used at the production site.
Security Group at DR Site	The security group that is used at the disaster recovery site.

After the security group mapping is created, it is displayed on the **Security Group Mapping** tab.

Overview	Prote	ction Group	Recovery Plan	Network & Security	Tasks				Edit Delete Site Pair
Zone Map		f hdr.Ebs	CdrSitePair.Infrastru	ctureTabltem.SecurityGro	upMapping.TipInfo				
vSwitch Ma		Add Secur	ity Group Mapping						
Security G		Production	Site Security Group		Security Group at DR Site	Status	Status Description	Operation	
		doctest-an sg-g	quanzu-a		doctest-anquanzu-a sgias	✓ Normal	The mapping status is normal.	Edit Delete	

If you want to edit or delete the created security group mapping, click Edit or Delete in the Operation column of the security group mapping.

6.3.1.2.4. Step 4: Create a protection group

This topic describes how to create a protection group.

Procedure

- 1. Log on to the Hybrid Disaster Recovery (HDR) console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click **Site Pairs**.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protection Group tab.
- 4. On the Protection Group tab, click **Create Protection Group**.
- 5. In the Create Protection Group dialog box, configure the parameters that are described in the following table and click OK.

Parameter	Description
Name	The name of the protection group. You can enter a descriptive name.
Expected RPO	The expected recovery point objective (RPO) of the protection group. An RPO is a time-based measurement of the maximum amount of data loss that is tolerable to a business system.
Protection group bandwidth settings	The allowed bandwidth of the protection group. Valid values: 80 to 640. Unit: Mbit/s. The bandwidth of each disk replication pair in the protection group is the same as that of the protection group and cannot be separately modified.
Production site save set	The default storage set in which disks are created at the production site. This storage set applies only to new disks. Existing disks at the production site are not affected.
Disaster recovery site save set	The default storage set in which disks are created at the disaster recovery site. This storage set applies only to new disks. Existing disks at the disaster recovery site are not affected.

After the protection group is created, it is displayed on the Protection Group tab.

Protection Group Name/ID	Monitoring	Protected Instances	Disaster Recovery Directio	Protection group bandwidth	Rpo 🛞	Status	Actions
doctest2 pg- x2	2	1	-	360Mbps	Expected RPO: 15 minutes Actual RPO: Recent RTO time:	Adding Protected Instance	Manage Protected Instance Enable Replication Disable Replication

If you want to modify or delete the created protection group, find the protection group, move the pointer over the More icon in the Actions column, and then choose Manage Protection Group > Edit or Manage Protection Group > Delete.

6.3.1.2.5. Step 5: Add instances to be protected

This topic describes how to add instances to be protected.

Procedure

- 1. Log on to the Hybrid Disaster Recovery (HDR) console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click **Site Pairs**.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protection Group tab.
- 4. On the Protection Group tab, find the protection group that you want to manage and click Manage Protected Instance in the Actions column.
- 5. On the Protected Instances page, click Add Protected Instance
- 6. In the Add Protected Instance panel, select one or more Elastic Compute Service (ECS) instances that you want to protect and click Next Step.

() Important

 After a secondary elastic network interface (ENI) is bound to an ECS instance, some images cannot automatically identify the IP address of the secondary ENI or add a route. As a result, the secondary ENI cannot work properly.

 If the value in the Secondary ENI column is Yes, an ECS instance is configured with a secondary ENI. In this case, check the IP address of the secondary ENI after a failover. This ensures that the secondary ENI works as expected. For more information, see the Configure a secondary ENI topic of the "Elastic Network Interfaces" chapter in Elastic Compute Service User Guide.

dd Pro	tected Instance	2								
	1	Select Instan	ce				2 C	onfirm Res	ources	
🚹 You d	an select up to 10 ECS	instances at a	time.							
	Instance ID/Name	St	atus	IP Address		Specifica	ations	Operating	System Seco	ondary ENI
	i- d	~ ~	Running	1)	2 vCPU	2 GB	linux	No	
	i- d	′ ~	Running	1)	2 vCPU	2 GB	linux	No	
CS instar	nces are selected, as sh	own in the follo	wing table:							
Instance	ID/Name	Status	IP Addr	ess	Specifica	tions	Operating S	System	Secondary ENI	Action
i- di	bv	🗸 Running	g 1	e)	2 vCPU 2	GB	linux		No	Remov

7. In the Confirm Resources step, confirm the basic information and resource information of the instances and click Confirm.

After the instances are added to the protection group, they are displayed on the Protected Instances page.

Protected Instance ID	ID/Name of Instance at Production Site	Instance Release F ID/Name of Instance at DR Site	Status	Replication Health Status	Actions
pi-0 :ei0	i-g vobv do	Op	✓ Initialized	-	Details Delete Protected Instance Change Instanc

What to do next

After a protected instance is added, you can perform the operations that are described in the following table.

Operation	Description
View details	Click Details to view the computing, network, and storage information of the protected instance.
Remove a protected instance	Remove the protected instance from the protection group. After the protected instance is removed, the instance is no longer protected by HDR.
Enable or disable instance release protection	If you turn on Instance Release Protection , the protected instance cannot be manually released in the console or by an API operation. Before you release the protected instance, you must disable instance release protection. Important Subscription ECS instances do not support instance release protection.
Change the instance type at the disaster recovery site	Change the ECS instance type of the protected instance at the disaster recovery site. Important Select an instance type based on your business requirements. The changed ECS instance type will be applied the next time you create an ECS instance. Existing ECS instances are not affected.

6.3.1.2.6. Step 6: Start replication

After you add protected instances for disaster recovery, you can start data replication to prepare for failover and failback.

Procedure

- 1. Log on to the Hybrid Disaster Recovery (HDR) console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click Site Pairs.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protection Group tab.
- 4. On the Protection Group tab, find the protection group that you want to manage and click Enable Replication in the Actions column.
- 5. In the Enable Replication panel, configure and confirm resources, and click Confirm.

! Important

If the instance type or operating system of an Elastic Compute Service (ECS) instance at the disaster recovery site does not meet your business requirements, you can follow the instructions in the console to change the instance type or operating system.

Enable Replication	n		
	Configuration	Confirm Resources	
Basic Information			
Region of DR Site	cn-w	Zone of DR Site cn-w	
Save set	Not enabled	Create Instance Total: 1. The instance types and ope systems are normal. View Details	rating
Cloud Disk 1 block, total 20 GB	cloud_pperf total	1 block, 20 GB in	
Instance 1	ecs.g6e-x25- c1m2.large	1	
Instance Cloud D	isk		
Instance Series		Instance Type	
ecs.g6e-x25		ecs.g6e-x25-c1m2.large	

After you start replication, the protection group enters the **Enabling Replication**, **Replicating Full Data**, and **Replicating Incremental Data** states in sequence.

Protection Group Name/ID	Monitoring	Protected Instances	Disaster Recovery Directio	Protection group bandwidth	Rpo 🗇	Status	Actions
dcotest pg- 5147	ы	1	-	360Mbps	Expected RPO: 15 minutes Actual RPO: Recent RTO time:	Enabling Replication	Manage Protected Instance Enable Replication Disable Replication

- Enabling Replication: HDR creates ECS instances at the disaster recovery site and starts data replication.
- Replicating Full Data: HDR is replicating valid data from protected instances to the disaster recovery site.
- Replicating Incremental Data: After full replication is complete, initial full data is replicated to the disaster recovery site. HDR continuously replicates incremental data from the production site to the disaster recovery site to meet recovery point objective (RPO) requirements.

6.3.1.2.7. Step 7: Perform a failover

After a protection group enters the Replicating Incremental Data state, you can perform a failover. This topic describes the basic operations of failover and failback.

Failover types

Switch After Data Synchronization

During the failover, Hybrid Disaster Recovery (HDR) stops the protected instances in the protection group, and performs the final data synchronization after all the protected instances are stopped. The failover starts after the data is synchronized. This ensures that the data at the disaster recovery site is the same as that at the production site. This type of failover applies to scenarios such as planned disaster recovery drills and business migration.

Switch Now

During the failover, HDR attempts to stop the protected instances in the protection group. HDR does not wait until all the protected instances are stopped or perform the final data synchronization. Some data may be lost within the recovery point objective (RPO) range. This type of failover applies to scenarios in which a fault cannot be rectified within a short period of time at the production site and business must be immediately switched to the disaster recovery site.

Failback types

Switch After Data Synchronization

During the failback, HDR stops the protected instances in the protection group, and performs the final data synchronization after all the protected instances are stopped. The failback starts after the data is synchronized. The service unavailability time is longer than the time for the immediate failback. The production site works properly in such failback scenarios.

Switch Now

During the failback, HDR attempts to stop the protected instances in the protection group. HDR does not wait until all the protected instances are stopped or perform the final data synchronization. HDR immediately initiates a failback, and some data may be lost. This type of failback applies to disaster recovery scenarios.

Switch After Data Synchronization

Forward protection: Perform a failover

- 1. Log on to the HDR console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click **Site Pairs**.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protection Group tab.
- 4. On the Protection Group tab, find the protection group that you want to manage, move the pointer over the i i i con in the Actions column, and then

choose Forward Protection > Failover.

- In the Start Failover dialog box, select Switch After Data Synchronization as Type and click OK. The protection group enters the Failover in Progress state.
- After the failover is complete, the state of the protection group changes to Failover Succeeded.
- Wait a few minutes until the failover is complete. Find the protection group and click Manage Protected Instance in the Actions column to go to the Protected Instances page.
- 7. Click each instance ID in the ID/Name of Instance at DR Site column to verify the data and applications on Elastic Compute Service (ECS) instances at the disaster recovery site.
- After the verification is complete, go back to the Protection Group tab. Find the protection group, move the pointer over the icon in the Actions column, and then choose Forward Protection > Failover Completed. In the message that appears, click OK.

The protection group enters the Failover Confirmed state. In this case, the protected instances at the production site are in the **Stopped** state, and the ECS instances at the disaster recovery site are in the **Running** state.

Enable reverse replication

Reverse replication is the process of replicating data from the disaster recovery site to the production site to prepare for failback.

- 1. Log on to the HDR console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click Site Pairs.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protection Group tab.
- 4. On the Protection Group tab, find the protection group that you want to manage and click Enable Reverse Replication in the Actions column.
- 5. In the Enable Reverse Replication panel, configure the Copy to Original Instance parameter, confirm the resources, and then click OK. The protection group enters the Reverse Replicating state.

() Important

The existing instances and disks at the disaster recovery site are used in the forward protection and failover. If the instance configurations at the disaster recovery site are modified, the configurations may be lost after forward protection is enabled.

Reverse protection: Perform a failback

After the fault at the production site is rectified, you must perform a failback to restore the data from the disaster recovery site to the production site. If the protection group is in the **Reverse Replicating** state and the value of **Actual RPO** meets your expectation, you can perform a failback.

1. Find the protection group that you want to manage, click the i fion in the Actions column, and then choose **Reverse Protection > Failback**. In the

dialog box that appears, select Switch After Data Synchronization and click OK.

The protection group enters the Failback in Progress state.

After the failback is complete, the protection group and its protected instances enter the Failback Completed state. In this case, the protected instances at the production site are in the **Running** state, and the ECS instances at the disaster recovery site are in the **Stopped** state.

2. After you confirm that the failback is complete, go back to the Protection Group tab. Find the protection group, move the pointer over the i con in

the Actions column, and then choose **Reverse Protection > Failback Completed**. In the message that appears, click **OK**. After the failback is complete, the state of the protection group changes to **Initializing**.

Switch Now

🔥 Warning

This type of failover or failback applies to scenarios in which serious faults occur on protected instances. During the failover or failback, HDR does not wait until all the protected instances at the production site is stopped or data replication is complete. Some data may be lost within the RPO range.

Forward protection: Perform a failover

- 1. Log on to the HDR console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click **Site Pairs**.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protection Group tab.
- 4. On the Protection Group tab, find the protection group that you want to manage, move the pointer over the i i con in the Actions column, and then

choose Forward Protection > Failover.

5. In the Start Failover dialog box, select Switch Now as Type and click OK.

] Important

and click **OK**.

If you perform this operation, some data may be lost. Proceed with caution.

The protection group enters the Failover in Progress state.

After the failover is complete, the state of the protection group changes to Failover Succeeded.

- Wait a few minutes until the failover is complete. Find the protection group and click Manage Protected Instances in the Actions column to go to the Protected Instances page appears.
- 7. Click each instance ID in the ID/Name of Instance at DR Site column to verify the data and applications on ECS instances at the disaster recovery site.
- 8. After the verification is complete, go back to the Protection Group tab. Find the protection group, move the pointer over the icon in the Actions

column, and then choose **Forward Protection > Failover Completed**. In the message that appears, click **OK**. The protection group enters the Failover Confirmed state. In this case, the protected instances at the production site are in the Stopped state, and the ECS instances at the disaster recovery site are in the Running state.

Reverse protection: Perform a failback

1. On the Protection Group tab, find the protection group that you want to manage, move the pointer over the i i con in the Actions column, and then

choose Reverse Protection > Enable Replication.

- In the Enable Reverse Replication panel, configure and confirm resources, and click OK. The protection group enters the Reverse Replicating Data state.
- 3. Find the protection group, move the pointer over the i i con in the Actions column, and then choose **Reverse Protection > Failback**. In the dialog

box that appears, select Switch Now as Type and click OK.

The protection group enters the Failback in Progress state.

() Important

If you perform this operation, some data may be lost. Proceed with caution.

After the failback is complete, the protection group and its protected instances enter the Failback Completed state. In this case, the protected instances at the production site are in the Running state, and the ECS instances at the disaster recovery site are in the Stopped state. 4. After the verification is complete, go back to the Protection Group tab, click the is in the Actions column of the protection group, and then

choose Reverse Protection > Failback. In the message that appears, click OK.

⑦ Note
After the failback is complete, the state of the protection group changes to Initializing. You can start forward replication again.

6.3.1.3. CDR

6.3.1.3.1. Limits

This topic describes the limits of continuous data replication (CDR), such as limits on operating systems, databases, and applications.

Operating systems

The following table describes the operating systems that are supported.

Operating system	Version
Windows Server	2008 R2, 2012, 2012 R2, and 2016

	() Important You must make sure that the /boot partition and the / partition reside on the same disk. If the partitions do not reside on the same disk, move the partitions to the same disk, and then register the Elastic Compute Service (ECS) instance for which you want to enable CDR.						
	Red Hat Enterprise Linux 7.0 to 7.9						
	Red Hat Enterprise Linux 6.0 to 6.10						
	CentOS 7.0 to 7.9						
	• CentOS 6.0 to 6.10						
	③ Note Only CentOS 64-bit is supported. If you need to use CentOS 6.x 32-bit, contact Alibaba Cloud for technical support.						
	SUSE Linux Enterprise Server 12.0 to 12.3						
Linux	 Important Only SUSE Linux Enterprise Server 64-bit is supported. If you need to use SUSE Linux Enterprise Server 12.x 32-bit, contact Alibaba Cloud for technical support. If SUSE Linux Enterprise Server 12.1 runs on a VMware VM, a black screen appears after you restart the VM. The black screen is caused by operating system errors, but not by Hybrid Disaster Recovery (HDR). 						
	 Alibaba Cloud Linux 2.1903 LTS 64-bit The following kernel versions of Alibaba Cloud Linux 2.1903 LTS 64-bit are supported: 4.19.91-25.1.al7.x86_64 4.19.91-23.al7.x86_64 4.19.91-22.2.al7.x86_64 						
	() Important If you need to use other kernel versions, contact Alibaba Cloud for technical support.						
	•						

Databases and applications

You can apply the replication technology of HDR to all types of databases and applications.

In most cases, automated scripts are required for various applications to ensure consistency among updates. You can use the tools and scripts that are provided by HDR to implement CDR. This ensures the smooth recovery of applications.

Other limits

HDR also has the following limits:

- If the size of a physical volume on which the system disk of an ECS instance resides exceeds 2 TB, you cannot restore full data on the ECS instance.
- A single physical volume on which a data disk resides cannot exceed 32 TB.
- Disk write limits:
- ∘ Linux

If the average I/O size is 4 KB, the maximum disk write speed is about 10 MB/s. If the average I/O size is 64 KB, the maximum disk write speed is about 30 MB/s.

Windows

The maximum disk write speed is 10 MB/s.

If the I/O size is smaller or the amount of data written to disks is larger, the recovery point objective (RPO) is not affected, but the recovery time objective (RTO) is extended. If the amount of data exceeds the disk write limit, both the RPO and RTO may be extended. When you design a disaster recovery solution, you must evaluate the business situation of protected instances and estimate the amount of data written to disks.

6.3.1.3.2. Step 1: Plan resources

Before you perform disaster recovery, you must plan the computing, storage, and network resources required for disaster recovery.

Computing resources

Make sure that the disks of Elastic Compute Service (ECS) instances to be protected meet the following requirements:

1. Disks are not encrypted.

2. The multi-attach feature is disabled.

Storage resources

To replicate data from the production site to the disaster recovery site, you must plan the required storage space in advance. Make sure that you have sufficient storage resources. If you need to use a storage set at the disaster recovery site, configure the storage set at the disaster recovery site in advance by referring to the storage set configuration manual.

Network resources

Create the virtual private clouds (VPCs), vSwitches, and security group rules required by the disaster recovery site.

6.3.1.3.3. Step 2: Create a site pair

Before you perform disaster recovery, you must create a site pair for centralized management of site resources. This topic describes how to create a site pair.

Prerequisites

A zone is selected to deploy the disaster recovery site. A virtual private cloud (VPC) is created in the zone. A vSwitch for replication and a vSwitch for restoration are created in the VPC.

Procedure

1. Log on to the Hybrid Disaster Recovery (HDR) console.

For more information, see Log on to the HDR console

- 2. In the left-side navigation pane, click Site Pairs.
- 3. On the Site Pairs page, click Create Site Pair.
- 4. In the Create Site Pair panel, configure the parameters that are described in the following table, and click OK.

Category	Parameter	Description
Disaster recovery type	Туре	The disaster recovery type. In this scenario, select Cross-zone Disaster Recovery. This replicates data from one zone to another zone in the specified region in real time.
Replication technology	Replication Technology	The type of technology used to replicate data for disaster recovery. In this scenario, select CDR . To use continuous data replication (CDR), you must install an agent on each protected instance. This replication technology is suitable for the scenario in which you want to implement disaster recovery for a single Elastic Compute Service (ECS) instance and can install software on the ECS instance. This technology has requirements for the operating systems of protected instances. You must check whether the operating systems of your instances are supported. For more information, see Limits.
Information of the production site and	Name	 The name of the production site or disaster recovery site. The name cannot exceed 60 characters in length. A recognizable name is required. Operations such as the deployment of a disaster recovery gateway must be performed after the site pair is created. The production site is used to specify the location of the instances that require disaster recovery on the cloud. The computing and storage resources that are used by the disaster recovery site are created in the specified VPC.
disaster recovery site	Region	The region in which the production site or disaster recovery site resides. Disaster recovery data is restored in the region.
	VPC	The VPC that is used by the production site or disaster recovery site.
	Zone	The zone in which the production site or disaster recovery site resides.

After the site pair is created, the site pair is displayed on the **Site Pairs** page.

If the site pair is incorrectly configured, you can click Delete to the right of the site pair to delete the site pair and then create a site pair again.

6.3.1.3.4. Step 3: Configure network and security settings

This topic describes the infrastructure configurations for disaster recovery, including vSwitch mappings and security group mappings.

Create a vSwitch mapping

The type of the current site pair is **Cross-zone Disaster Recovery**, so you do not need to create a zone mapping. To create a vSwitch mapping, perform the following steps:

- 1. Log on to the Hybrid Disaster Recovery (HDR) console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click Site Pairs.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Network & Security tab.
- 4. In the left sidebar, click vSwitch Mapping, and click Add vSwitch Mapping.
- 5. In the Add vSwitch Mapping dialog box, configure the parameters that are described in the following table, and click OK.

() Important

If the CIDR block of the vSwitch at the production site is different from that at the disaster recovery site, IP address inconsistency occurs when you perform a failover or failback.

Parameter	Description					
Network of Production Site	The CIDR block of the vSwitch at the production site. Note For example, if the CIDR block of the vSwitch at the production site is 192.168.0.0/24 and the CIDR block of the vSwitch at the disaster recovery site is 192.168.1.0/24, IP addresses are mapped based on the following rule: 192.168.0.1 mapped to 192.168.1.1.					
Network of DR Site	The CIDR block of the vSwitch at the disaster recovery site.					

After the vSwitch mapping is created, it is displayed on the vSwitch Mapping tab.

Zone Map…	hdr.EbsCdrSitePair.InfrastructureTabItem.Networl	cMapping.TipInfo				
vSwitch M…	Zone Mapping1 cn-wu	1-a				Add vSwitch Mapping
Security Gr	Network of Production Site	Network of DR Site	Status	Status Description		
	do vsv 0/24	dc vs 1.0/24	✓ Normal	The mapping status is normal.	Edit Delete	

If you want to edit or delete the created vSwitch mapping, click Edit or Delete on the right side of the vSwitch mapping.

Create a security group mapping

1. Log on to the HDR console.

- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click **Site Pairs**.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Network & Security tab.
- 4. In the left sidebar, click Security Group Mapping, and click Add Security Group Mapping.
- 5. In the Add Security Group Mapping dialog box, configure the parameters that are described in the following table, and click OK.

Parameter	Description
Production Site Security Group	The security group that is used at the production site.
Security Group at DR Site	The security group that is used at the disaster recovery site.

After the security group mapping is created, it is displayed on the **Security Group Mapping** tab.

Overview	Protec	cted Instance	Network & Security	Tasks							Edit	Delete Site Pair
Zone Map…		🚯 hdr. EbsCadSitePair.InfrastructureTablitem. SecurityGroupMapping.TipInfo										
vSwitch Ma		Add Securi	ly Group Mapping									
Security G		Production	Site Security Group	Security (Group at DR Site	:	Status		Status Description	Operation		
		d si	/1qh	dc sg	qh		✓ Normal		The mapping status is normal.	Edit Delete		

If you want to edit or delete the created security group mapping, click Edit or Delete in the Operation column of the security group mapping.

6.3.1.3.5. Step 4: Add instances to be protected

This topic describes how to add instances to be protected.

Procedure

- 1. Log on to the Hybrid Disaster Recovery (HDR) console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click **Site Pairs**.
- 3. On the **Site Pairs** page, click the ID of the site pair that you want to manage.
- 4. On the page that appears, click the Protected Instance tab. Then, click Add Protected Instance.
- 5. In the Add Protected Instance panel, select one or more Elastic Compute Service (ECS) instances that you want to protect, configure and confirm resources, and then click Confirm.

() Important

- After a secondary elastic network interface (ENI) is bound to an ECS instance, some images cannot automatically identify the IP address of the secondary ENI or add a route. As a result, the secondary ENI cannot work properly.
- If the value in the Secondary ENI column is Yes, an ECS instance is configured with a secondary ENI. In this case, check the IP address of the secondary ENI after a failover. This ensures that the secondary ENI works as expected. For more information, see the Configure a secondary ENI topic of the "Elastic Network Interfaces" chapter in Elastic Compute Service User Guide.

After the instances are added, they are displayed on the Protected Instance tab. The instances are in the Initialized state.

What to do next

After a protected instance is added, you can perform the operations that are described in the following table.

Operation	Description
Install an agent	Install a disaster recovery agent for the protected instance. If the agent fails to be installed, you can perform this operation to install it again.
Restart a protected instance	Restart the protected instance. For example, after a disaster recovery agent is installed on a Windows protected instance, you must restart the instance.
Remove a protected instance	Remove the protected instance.

Enable instance release protection	If you turn on Release protection , the protected instance cannot be manually released in the console or by an API operation. Before you release the protected instance, you must disable instance release protection.
Disable instance release protection	If you turn off Release protection , you can directly release the protected instance.
View details	 View the computing information of the protected instance, including resource information and resource configurations. For example, view the ECS instance ID and instance type. View the network information of the protected instance, including ENI information and ENI configurations. For example, view the ENI ID and IP address. View the storage information of the protected instance, including disk information and disk configurations. For example, view the disk name and type.

6.3.1.3.6. Step 5: Start replication

After you add protected instances for disaster recovery, you can start data replication to prepare for failover and failback.

Procedure

- 1. Log on to the Hybrid Disaster Recovery (HDR) console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click **Site Pairs**.
- 3. On the **Site Pairs** page, click the ID of the site pair that you want to manage. On the page that appears, click the Protected Instance tab. Then, find the protected instance that you want to manage, move the pointer over the pictor in the Operation column, and then choose **Forward Protection**

> Start Replication.

4. In the Enable Replication panel, configure the parameters that are described in the following table, confirm resources, and then click Confirm.

Parameter	Description	
Recovery Point Policy	The policy for creating recovery points. HDR creates a recovery point at the specified interval in units of hours every day.	
Automatic restart after replication interruption	Specifies whether to automatically fix replication interruption on the Linux instance or restart the Windows instance to resume replication if replication is interrupted.	
Hard Disk Type	The disk type of the shadow instance. The following disk types are supported: standard performance disk, SSD, ultra disk, and premium performance disk.	
Replication Network	The vSwitch that HDR uses to replicate data for disaster recovery. By default, HDR reads the available vSwitches of the virtual private cloud (VPC) in which the disaster recovery site resides. If the vSwitch used for replication and the vSwitch at the disaster recovery site are not in the same zone, the recovery time objective (RTO) is extended.	
	Important When replication is started, a shadow instance is created and randomly occupies an IP address of the vSwitch used for replication. We recommend that you separate the CIDR block of the vSwitch used for replication from the CIDR block of the vSwitch at the disaster recovery site. This prevents the shadow instance from occupying the IP addresses of Elastic Compute Service (ECS) instances at the disaster recovery site.	
Use a Save set	Specifies whether to use a storage set at the disaster recovery site. If you turn on Use a Save set , configure the storage set at the disaster recovery site in advance by referring to the storage set configuration manual.	

After you start replication, the protected instance enters the Enabling Replication, Replicating Full Data, and Replicating states in sequence.

- $\circ~$ **Enabling Replication**: HDR is ready to start replication.
- Replicating Full Data: HDR is replicating valid data from the protected instance to the disaster recovery site.
- **Replicating**: After full replication is complete, full data is replicated to the disaster recovery site. Then, HDR monitors all write operations on the disks of the protected instance and continuously replicates incremental data to the disaster recovery site.

6.3.1.3.7. Step 6: Perform a failover

After a protected instance enters the Replicating state, you can perform a failover.

Usage notes

During the failover, an Elastic Compute Service (ECS) instance is created, and storage and network resources are created or migrated. To prevent network resource conflicts, you can perform one of the following operations:

- Make sure that other protected instances are not affected by modifying the network and security group mappings in the infrastructure.
- · Clear the ECS instance at the disaster recovery site.
- Modify the private IP address of the ECS instance at the disaster recovery site.

After a protected instance is added to a protection group that uses the continuous data replication (CDR) and before replication is started, data disks that are newly attached to the ECS instance are automatically added to the protection group after a period of time if the specifications of the protection group allow. Detached data disks are automatically removed from the protection group. In other cases, an alert is generated if you attach data disks to or detach data disks from the ECS instance. This does not affect disaster recovery for disks that have been added to the protection group but may affect disaster recovery for the overall business. We recommend that you stop the protection group, change the configurations of the ECS instance, and then add the ECS instance to the protection group again.

(Optional) Perform a disaster recovery drill

After a protected instance enters the Replicating state, you can perform a disaster recovery drill on the instance.

A disaster recovery drill is an important part of disaster recovery. A disaster recovery drill allows you to run a protected ECS instance on the cloud to verify the correctness of relevant applications. A disaster recovery drill has the following core features:

• Allows you to easily check whether an application can run on a restored ECS instance as expected.

• Familiarizes you with the disaster recovery process and ensures that a smooth failover can be performed if the production site encounters a failure.

? Note

When you initiate a disaster recovery drill, Hybrid Disaster Recovery (HDR) creates a drill VPC at the disaster recovery site. All ECS instances that are created during the drill are created in this drill VPC. The instances in the disaster recovery VPC are not affected. When you clear the drill environment, the drill VPC is deleted.

To perform a disaster recovery drill, perform the following steps:

1. Log on to the HDR console.

For more information, see Log on to the HDR console.

- 2. In the left-side navigation pane, click Site Pairs.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protected Instance tab.
- 4. Find the protected instance for which you want to perform a disaster recovery drill and click **Test Failover** in the Operation column.
- 5. In the Test Failover panel, configure the parameters that are described in the following table, and click OK.

Parameter	Description
Instance Name	The name of the ECS instance that is automatically generated for disaster recovery. By default, this parameter cannot be modified.
Recovery Point	Select a recovery point from the Recovery Point drop-down list. By default, the ECS instance is created by using the data at the current point in time.

HDR creates an ECS instance by using the data of the specified recovery point. During the disaster recovery drill, real-time data replication is not affected.

6. Wait until the disaster recovery drill is complete, which takes several minutes. Click the link in the **ID/Name of Instance for DR Drill** column to verify data and applications on the restored ECS instance.

7. After the verification is complete, click Cleanup Test Environment in the Operation column. The restored ECS instance is deleted.

? Note

After you verify data and applications on the ECS instance that is created during the disaster recovery drill, we recommend that you clear the drill environment at the earliest opportunity to reduce resource usage.

Perform a failover

Regular disaster recovery drills ensure that you can run your applications on restored ECS instances at any time. If a critical fault occurs at the production site and you need to immediately restart your core business on the cloud, you must perform a failover.

- ▲ Warning
 - The failover operation is applicable if a protected instance encounters a critical fault. If you perform this operation, real-time data replication is stopped. You must restart the replication and complete a full replication to continue disaster recovery protection for the protected instance.
 - Before a failover, clear the drill environment. Wait a few minutes and then start the failover.
- 1. Log on to the HDR console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click Site Pairs.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protected Instance tab.
- 4. Configure the instance that requires forward protection or reverse protection.

On the **Protected Instance** tab, find the instance that you want to configure and click **Configuration** in the Operation column. In the **Instance Configuration** panel, configure the parameters that are described in the following table, and click **OK**.

Parameter	Description	
Copy Direction	The replication direction. Valid values: Forward Protection and Reverse Protection.	
Instance Name	The name of the ECS instance that is automatically generated for disaster recovery. By default, this parameter cannot be modified.	
ECS Instance Type	The instance type of the instance at the disaster recovery site. The system automatically selects the same instance type as that of the instance at the production site.	

Hard Disk Type	The type of disks that are used by the restored instance. Valid values: Common performance cloud disk SSD disk Efficient cloud disk High-Performance Cloud Disk 	
Use a Save set	If you turn on Use a Save set, you must configure the StorageSet Name and Partition parameters.	
Post Script	The script you want to execute after the failover.	

5. In the Operation column of the protected instance, click the i licon and choose Forward Protection > Failover.

6. In the **Failover** panel, confirm the instance name and recovery point, and click **OK**.

The protected instance enters the Failover in Progress state.

() Important

You can restore the ECS instance to the current point in time only once. By default, the ECS instance is created by using the data at the current point in time.

After the failover is complete, you can click the link in the ID/Name of Instance at DR Site column to verify data and applications.

If the applications are running as expected at the current point in time, click the i icon and choose Forward Protection > Commit Failover to commit the failover. If you commit the failover, all recovery points are deleted, and you cannot change the recovery point of the ECS instance.

⑦ Note

After the failover or recovery point change is complete, and you confirm that the applications on the restored ECS instance have taken over the business, you can commit the failover. This clears the resources that are occupied by disaster recovery replication on the cloud to reduce resource usage.

• The applications at the current point in time may not work as expected. For example, database consistency issues may occur, or the contaminated source data has been synchronized to another region. In this case, click the i icon, choose Forward Protection > Change Recovery Point, and

then select another recovery point.

Note
 The process of changing the recovery point is similar to the failover process. You need to only select an appropriate recovery point.

Perform a reverse replication

After the applications on a protected instance are replicated from one zone (such as Zone A) to another zone (such as Zone B), you can perform a reverse replication to replicate the applications from Zone B to Zone A.

1. Log on to the HDR console.

For more information, see Log on to the HDR console.

- 2. In the left-side navigation pane, click Site Pairs.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protected Instance tab.
- 4. Find the protected instance that you want to manage, click the i con in the Operation column, and then choose Reverse Protection > Initiate

Reverse Replication.

5. In the Initiate Reverse Replication dialog box, configure the parameters that are described in the following table, and click OK.

? Note

If you do not restore data to the original ECS instance, HDR creates another ECS instance and uses new storage and network resources. The ECS instance at the original disaster recovery site and its storage and network resources are all retained. To prevent network resource conflicts, you can perform one of the following operations:

i. Make sure that other protected instances are not affected by modifying the network and security group mappings in the infrastructure.

ii. Clear the ECS instance at the production site.

iii. Modify the private IP address of the ECS instance at the production site.

Parameter	Description
Crash Consistent Point Policy	The policy for creating recovery points. HDR creates a recovery point at the specified interval in units of hours every day.
Whether to restore to the original ECS	Specifies whether to restore data to the original ECS instance. If you select this check box, the original instance is used as a shadow instance and the data of the original instance is deleted. If you do not select this check box, you must configure the ECS specifications such as the instance type and disk type as prompted.

The protected instance enters the Reverse Registering, Reverse Initialized, Enabling Reverse Replication, Reverse Replicating Full Data, and Reverse Replicating states in sequence. The replication progress is displayed in real time.

6. After the protected instance enters the Reverse Replicating state, start a failback.

i. In the Operation column of the protected instance, click the isi icon and choose **Reverse Protection > Failback**.

ii. In the Failback panel, configure the parameters that are described in the following table, and click OK.

() Important

During the failback, an ECS instance is created, and storage and network resources are created or migrated. To prevent network resource conflicts, you can perform one of the following operations:

- Make sure that other protected instances are not affected by modifying the network and security group mappings in the infrastructure.
 Clear the ECS instance at the production site.
- Clear the ECS instance at the production site.
- Modify the private IP address of the ECS instance at the production site.

Parameter	Description
Instance Name	The name of the ECS instance that is automatically generated for disaster recovery. By default, this parameter cannot be modified.
Recovery Point	Select a recovery point from the Recovery Point drop-down list. By default, the ECS instance is created by using the data at the current point in time.

The protected instance enters the Failback in Progress state. After the failback is complete, you can click the link in the **ID/Name of Instance at Production Site** column to check the status of the restored ECS instance.

6.3.2. Cross-cloud disaster recovery

6.3.2.1. Cross-cloud disaster recovery

Cross-cloud disaster recovery indicates disaster recovery for Elastic Compute Service (ECS) instances between the production site and the disaster recovery site that are deployed on different clouds.

Solution description

The production site and disaster recovery site are deployed on two clouds. After cloud organization mapping is configured and a protection group is created, a channel between the production site and the disaster recovery site is established. The channel is used to replicate data from the production site to the disaster recovery site in real time. If the production site fails, services at the production site are failed over to the ECS instances at the disaster recovery site. After the production site is recovered, the services are failed back to the ECS instances at the production site. This ensures business continuity.

- 1. Contact an O&M administrator to log on to the Apsara Uni-manager Operations Console and create a cloud site in Hybrid Disaster Recovery (HDR) Ops. For more information, see Cloud Defined Storage O&M Guide.
- Log on to the Apsara Uni-manager Management Console, go to the HDR console, and then configure the mapping between the organizations of the production site and disaster recovery site. For more information, see Cloud configuration.

3. Perform disaster recovery operations in the HDR console.

Procedure

To implement cross-cloud disaster recovery for ECS instances, perform the following steps:

- 1. Configure cloud information
- Configure the mapping between the organizations of the production site and disaster recovery site. For more information, see Cloud configuration. 2. Step 1: Plan resources
- 2. Step 1: Plan resources

Before you implement disaster recovery, you must determine the region and zone in which you want to create a disaster recovery site. Create a virtual private cloud (VPC) and vSwitches at the disaster recovery site.

- 3. Step 2: Create a site pair Configure the CIDR blocks to be used at the disaster recovery site. During the test, you can use the default configurations to create a VPC and vSwitches at the disaster recovery site. You can also configure the same VPC CIDR block and vSwitch CIDR block for the production site and disaster recovery site. During actual disaster recovery, you can configure CIDR blocks as required.
- 4. Step 3: Configure network and security settings
- Map resources, including vSwitches and security groups.
- 5. Step 4: Create a protection group
 - Create a protection group.
- 6. Step 5: Add instances to be protected
- Add instances to be protected to the protection group.
- 7. Step 6: Start replication

Start disaster recovery protection. Data is replicated from the production site to the disaster recovery site.

- 8. Step 7: Perform a failover
 - Switch After Data Synchronization

During the failover, HDR stops the protected instances in the protection group, and performs the final data synchronization after all the protected instances are stopped. The failover starts after the data is synchronized. This ensures that the data at the disaster recovery site is the same as that at the production site. This type of failover applies to scenarios such as planned disaster recovery drills and business migration.

• Switch Now

During the failover, HDR attempts to stop the protected instances in the protection group. HDR does not wait until all the protected instances are stopped or perform the final data synchronization. Some data may be lost within the recovery point objective (RPO) range. This type of failover applies to scenarios in which a fault cannot be rectified within a short period of time at the production site and business must be immediately switched to the disaster recovery site.

6.3.2.2. Step 1: Plan resources

Before you implement cross-cloud disaster recovery, you must plan the computing, storage, and network resources required for disaster recovery, and contact an O&M administrator to configure the information of the peer cloud.

Background information

Based on the computing, storage, and network resources used by your business, plan and deploy resources in the data center at the disaster recovery

Configure cloud site information

Contact an O&M administrator to log on to the Apsara Uni-manager Operations Console and create a cloud site in HDR-Ops. For more information, see Cloud Defined Storage O&M Guide.

Computing resources

The disks on Elastic Compute Service (ECS) instances that require disaster recovery must meet the following requirements:

- 1. The disks are premium performance disks.
- 2. The disks are not encrypted.
- 3. The disks are not attached to multiple ECS instances.

Storage resources

To replicate data from the production site to the disaster recovery site, you must plan the required storage space in advance. Make sure that you have sufficient storage resources. If you need to use a storage set at the disaster recovery site, configure the storage set at the disaster recovery site in advance by referring to the storage set configuration manual.

Network resources

Create the virtual private cloud (VPC), vSwitches, and security group rules required by the disaster recovery site.

6.3.2.3. Step 2: Create a site pair

Before you perform disaster recovery, you must create a site pair for centralized management of site resources. This topic describes how to create a site pair.

Prerequisites

vSwitches are created at the disaster recovery site. The disaster recovery site supports vSwitches that reside in the same virtual private cloud (VPC) as the production site or a different VPC from the production site.

Limits

Hybrid Disaster Recovery (HDR) imposes limits on Elastic Compute Service (ECS) instances that require disaster recovery. For more information, see Limits.

Procedure

- 1. Log on to the HDR console.
- For more information, see Log on to the HDR console
- 2. In the left-side navigation pane, click **Site Pairs**.
- 3. In the upper-left corner of the Site Pairs page, click Create Site Pair.
- 4. In the Create Site Pair panel, configure the parameters that are described in the following table, and click OK.

Туре	Parameter	Description
Disaster recovery	Туре	 The type of disaster recovery. In this example, Cloud to Cloud is selected. Valid values: Cross-zone Disaster Recovery: asynchronously replicates data from one zone to another zone in the primary region. Cloud to Cloud: asynchronously replicates data from the production site to a disaster recovery site.
Replication technologies	Replication Technology	The type of technology used to replicate data for disaster recovery. In this scenario, Async Replication is selected. The async replication technology is implemented at the virtualization layer without the need to install an agent on the protected instance. This option applies to consistent disaster recovery on VM groups. The target customers are those who can accept a recovery point objective (RPO) of a few minutes and do not expect intrusions into the system.
Resource set	The Organization's resource set	The resource set to which the local cloud site belongs. HDR automatically uses the resource sets specified in the organization mapping to manage computing, network, and storage resources based on the cloud site configurations of the production and disaster recovery sites.
Information of the production site and	Cloud ID	The Alibaba Cloud instance ID. This parameter is displayed only if you set the Type parameter to Cloud to Cloud.
	Name	 The name of the production or disaster recovery site. Specify a descriptive name for easy identification. The name cannot exceed 60 characters in length. The production site is used to specify the location of the instances that require disaster recovery on the cloud. The computing and storage resources that are used by the disaster recovery site are created in the specified VPC.
disaster recovery site		
Region	The region in which the production or disaster recovery site resides. Disaster recovery data is restored in the region.	
--------	---	
VPC	The VPC used by the production or disaster recovery site.	
Zone	The zone in which the production or disaster recovery site resides.	

After the site pair is created, it is displayed on the **Site Pairs** page.

Site Pair ID	Site Pair Name		Site Pair Type	Replication Technology	Number of Protected Instances	Created At	Actions
s-00	docte	heb	Cross-zone Disaster Recovery	Async Replication	1	2023-12-27 11:16:38	Manage Site Pair Delete
s-00000 bz4	doctes	E-a	Cloud to Cloud	Async Replication	0	2023-06-27 16:02:40	Manage Site Pair Delete

If the site pair is incorrectly configured, you can click Delete to the right of the site pair to delete the site pair and then create a site pair again.

6.3.2.4. Step 3: Configure network and security settings

This topic describes how to configure the network and security settings required for disaster recovery, including vSwitch mappings and security group mappings.

Configure a vSwitch mapping

- To configure a vSwitch mapping, perform the following steps:
- 1. Log on to the Hybrid Disaster Recovery (HDR) console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click **Site Pairs**.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Network & Security tab.
- 4. In the left-side pane, click vSwitch Mapping. Then, click Add vSwitch Mapping.
- 5. In the Add vSwitch Mapping dialog box, configure the parameters that are described in the following table and click OK.

Parameter	Description
Network of Production Site	 The CIDR block of the vSwitch at the production site. Note For example, if the CIDR block of the vSwitch at the production site is 192.168.0.0/24 and the CIDR block of the vSwitch at the disaster recovery site is 192.168.1.0/24, IP addresses are mapped based on the following rule: 192.168.0.1 mapped to 192.168.1.1. If the CIDR block of the vSwitch at the production site is inconsistent with the CIDR block of the vSwitch at the production site is inconsistent with the CIDR block of the vSwitch at the P addresses that you use may be inconsistent when you perform a failover or failback.
Network of DR Site	The CIDR block of the vSwitch at the disaster recovery site.

After the vSwitch mapping is configured, it is displayed on the vSwitch Mapping tab.

If you want to modify or delete the configured vSwitch mapping, find the vSwitch mapping and click Edit or Delete.

Configure a security group mapping

- 1. Log on to the HDR console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click **Site Pairs**.
- 3. On the **Site Pairs** page, click the ID of the site pair that you want to manage. On the page that appears, click the **Network & Security** tab.
- 4. In the left-side pane, click Security Group Mapping. Then, click Add Security Group Mapping.
- 5. In the **Add Security Group Mapping** dialog box, configure the parameters that are described in the following table and click **OK**.

Parameter	Description
Production Site Security Group	The security group that is used at the production site.
Automatically Create Security Group at DR Site	Specifies whether to automatically create a security group at the disaster recovery site. If you turn on Automatically Create Security Group at DR Site , security group rules whose authorization objects are CIDR blocks are replicated from the production site to the disaster recovery site. If the system cannot obtain mappings from security group rules with other authorization objects, check whether the security group mapping that you want to configure is valid and try again.
Security Group at DR Site	The security group that is used at the disaster recovery site.

After the security group mapping is configured, it is displayed on the Security Group Mapping tab.

If you want to modify or delete the configured security group mapping, find the security group mapping and click Edit or Delete in the Operation column.

6.3.2.5. Step 4: Create a protection group

This topic describes how to create a protection group.

Procedure

- 1. Log on to the Hybrid Disaster Recovery (HDR) console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click **Site Pairs**.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protection Group tab.
- 4. On the Protection Group tab, click Create Protection Group.
- 5. In the Create Protection Group dialog box, configure the parameters that are described in the following table, and click OK.

Parameter	Description
Name	The name of the protection group. You can enter a descriptive name.
Expected RPO	The expected recovery point objective (RPO) of the protection group. An RPO is a time-based measurement of the maximum amount of data loss that is tolerable to a business system.
Zone of Production Site	The zone in which the production site resides.
Zone of DR Site	The zone in which the disaster recovery site resides.
Protection group bandwidth settings	The allowed bandwidth of the protection group. Valid values: 80 to 640. Unit: Mbit/s. The bandwidth of each disk replication pair in the protection group is the same as that of the protection group and cannot be separately modified.
Production site save set	The default storage set in which disks are created at the production site. This storage set applies only to new disks. Existing disks at the production site are not affected.
Disaster recovery site save set	The default storage set in which disks are created at the disaster recovery site. This storage set applies only to new disks. Existing disks at the disaster recovery site are not affected.

After the protection group is created, the protection group is displayed on the **Protection Group** tab. The **Expected RPO** value is displayed, and the protection group is in the **Initializing** state.

If the protection group is incorrectly configured or in an abnormal state such as Replication Error, you can clear the protection group, delete the protection group, and then create another protection group.

i. In the Actions column of the protection group, choose More > Clear Protection Group.

- ii. In the Clear Protection Group dialog box, enter the description and click OK to submit a ticket. Your ticket must be confirmed by an O&M administrator in the Apsara Uni-manager Operations Console.
- iii. Contact an O&M administrator to log on to the Apsara Uni-manager Operations Console and confirm the ticket in HDR Ops. Then, the protection group is cleared.

6.3.2.6. Step 5: Add instances to be protected

This topic describes how to add instances to be protected.

Procedure

1. Log on to the Hybrid Disaster Recovery (HDR) console.

For more information, see Log on to the HDR console.

- 2. In the left-side navigation pane, click Site Pairs.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protection Group tab.
- 4. On the Protection Group tab, find the protection group that you want to manage and click Manage Protected Instance in the Actions column.
- 5. On the **Protected Instances** page, click **Add Protected Instance**.

6. In the Add Protected Instance panel, select one or more Elastic Compute Service (ECS) instances that you want to protect and click Next Step.

() Important

- After an ECS instance is bound to a secondary elastic network interface (ENI), some images cannot automatically identify the IP address of the secondary ENI or add a route. As a result, the secondary ENI cannot work properly.
- If the value in the Secondary ENI column is Yes, an ECS instance is configured with a secondary ENI. In this case, check the IP address of the secondary ENI after a failover. This ensures that the secondary ENI works as expected. For more information, see the Configure a secondary ENI topic of the "Elastic Network Interfaces" chapter in Elastic Compute Service User Guide.

7. In the Confirm Resources step, confirm the inheritance information and resource details, and click Confirm.

After the instances are added to the protection group, they are displayed on the Protected Instances page.

Protected Instance ID	ID/Name of Instance at Production Site	Instance Release F	ID/Name of Instance at DR Site	Status	Replication Health Status	Actions
pi-0 tei0	i-g /obv do	Op		✓ Initialized	-	Details Delete Protected Instance Change Instance

More operations

After a protected instance is added, you can perform the operations that are described in the following table.

Operation	Description
Remove a protected instance	Remove the protected instance.
Enable instance release protection	If you turn on Release protection , the protected instance cannot be manually released in the console or by calling an API operation. Before you release the protected instance, you must disable instance release protection.
Disable instance release protection	If you turn off Release protection , you can directly release the protected instance.
	View the computing information of the protected instance, including resource information and resource configurations. For example, view the ECS instance ID and instance type.
View details	 View the network information of the protected instance, including ENI information and ENI configurations. For example, view the ENI ID and IP address.
	 View the storage information of the protected instance, including disk information and disk configurations. For example, view the disk name and type.
	Change the ECS instance type of the protected instance at the disaster recovery site.
Change the instance type at the disaster recovery site	① Important Select an instance type based on your business requirements. The changed ECS instance type will be applied the next time you create an ECS instance. Existing ECS instances are not affected.

What to do next

Step 6: Start replication

6.3.2.7. Step 6: Start replication

After you add a protected instance for disaster recovery, you can start data replication to prepare for failover and failback.

Procedure

- 1. Log on to the Hybrid Disaster Recovery (HDR) console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click Site Pairs.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protection Group tab.
- 4. Find the protection group that you want to manage and click Enable Replication in the Actions column.
- 5. In the Enable Replication dialog box, confirm the configurations and resources, and click OK.

() Important

If the instance type or operating system of the instance at the disaster recovery site does not meet your business requirements, you can follow the on-screen instructions to change the instance type or operating system.

After you start replication, the protection group enters the **Enabling Replication**, **Replicating Full Data**, and **Replicating Incremental Data** states in sequence.

- Enabling Replication: HDR creates a disaster recovery instance at the disaster recovery site and starts data replication.
- Replicating Full Data: HDR replicates all valid data on the instance at the production site to the disaster recovery site.
- Replicating Incremental Data: After full replication is complete, initial full data is replicated to the disaster recovery site. HDR continuously replicates incremental data from the production site to the disaster recovery site to meet recovery point objective (RPO) requirements.

6.3.2.8. Step 7: Perform a failover

After a protection group enters the Replicating Incremental Data state, you can perform a failover. This topic describes the basic operations of failover and failback.

Failover types

Switch After Data Synchronization

During the failover, Hybrid Disaster Recovery (HDR) stops the protected instances in the protection group, and performs the final data synchronization after all the protected instances are stopped. The failover starts after the data is synchronized. This ensures that the data at the disaster recovery site is the same as that at the production site. This type of failover applies to scenarios such as planned disaster recovery drills and business migration.

Switch Now

During the failover, HDR attempts to stop the protected instances in the protection group. HDR does not wait until all the protected instances are stopped or perform the final data synchronization. Some data may be lost within the recovery point objective (RPO) range. This type of failover applies to scenarios in which a fault cannot be rectified within a short period of time at the production site and business must be immediately switched to the disaster recovery site.

Failback types

• Switch After Data Synchronization

During the failback, HDR stops the protected instances in the protection group, and performs the final data synchronization after all the protected instances are stopped. The failback starts after the data is synchronized. The service downtime of this failback type is longer than that of the Switch Now type. This type of failback applies to scenarios in which the production site works properly.

Switch Now

During the failback, HDR attempts to stop the protected instances in the protection group. HDR does not wait until all the protected instances are stopped or perform the final data synchronization. HDR immediately initiates a failback, and some data may be lost. This type of failback applies to disaster recovery scenarios.

Perform a failover and a failback of the Switch After Data Synchronization type

Forward protection: Perform a failover

- 1. Log on to the HDR console.
- For more information, see Log on to the HDR console
- 2. In the left-side navigation pane, click **Site Pairs**.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protection Group tab.

choose Forward Protection > Failover.

- In the Start Failover dialog box, select Switch After Data Synchronization as Type and click OK. The protection group enters the Failover in Progress state.
- After the failover is complete, the state of the protection group changes to Failover Succeeded.
- Wait a few minutes until the failover is complete. Find the protection group and click Manage Protected Instance in the Actions column to go to the Protected Instances page.
- 7. Click each instance ID in the ID/Name of Instance at DR Site column to verify the data and applications on Elastic Compute Service (ECS) instances at the disaster recovery site.
- After the verification is complete, go back to the Protection Group tab. Find the protection group, move the pointer over the icon in the Actions column, and then choose Forward Protection > Failover Completed. In the message that appears, click OK.

The protection group enters the Failover Confirmed state. In this case, the protected instances at the production site are in the **Stopped** state, and the ECS instances at the disaster recovery site are in the **Running** state.

Start reverse replication

Reverse replication is the process of replicating data from the disaster recovery site to the production site to prepare for failback.

- 1. Log on to the HDR console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click Site Pairs.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protection Group tab.
- 4. On the Protection Group tab, find the protection group that you want to manage and click Enable Reverse Replication in the Actions column.
- 5. In the Enable Reverse Replication panel, configure the Copy to Original Instance parameter, confirm resources, and then click Confirm. The protection group enters the Reverse Replicating Data state.

() Important

The existing ECS instances and disks at the disaster recovery site are used in the forward protection and failover. If the instance configurations at the disaster recovery site are modified, the configurations may be lost after forward protection is enabled.

Reverse protection: Perform a failback

After the fault at the production site is rectified, you must perform a failback to restore the data from the disaster recovery site to the production site. If the protection group is in the **Reverse Replicating Data** state and the value of the **Actual RPO** parameter meets your expectation, you can perform a failback.

1. On the Protection Group tab, find the protection group that you want to manage, move the pointer over the i icon in the Actions column, and then

choose **Reverse Protection > Failback**. In the dialog box that appears, select **Switch After Data Synchronization** as Type and click **OK**. The protection group enters the **Failback in Progress** state.

After the failback is complete, the protection group and its protected instances enter the Failback Completed state. In this case, the protected instances at the production site are in the **Running** state, and the ECS instances at the disaster recovery site are in the **Stopped** state.

2. After you confirm that the failback is complete, go back to the Protection Group tab. Find the protection group, move the pointer over the i con in

the Actions column, and then choose **Reverse Protection > Failback Completed**. In the message that appears, click **OK**. After the failback is complete, the state of the protection group changes to **Initializing**.

Perform a failover and a failback of the Switch Now type

\land Warning

This type of failover or failback applies to scenarios in which serious faults occur on protected instances. During the failover or failback, HDR does not wait until all the protected instances at the production site are stopped or data replication is complete. Some data may be lost within the RPO range.

Forward protection: Perform a failover

- 1. Log on to the HDR console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click **Site Pairs**.
- 3. On the Site Pairs page, click the ID of the site pair that you want to manage. On the page that appears, click the Protection Group tab.
- 4. On the Protection Group tab, find the protection group that you want to manage, move the pointer over the i icon in the Actions column, and then choose **Forward Protection > Failover**.

5. In the Start Failover dialog box, select Switch Now as Type and click OK.

() Important

If you perform this operation, some data may be lost. Proceed with caution.

The protection group enters the Failover in Progress state.

After the failover is complete, the state of the protection group changes to Failover Succeeded.

- Wait a few minutes until the failover is complete. Find the protection group and click Manage Protected Instance in the Actions column to go to the Protected Instances page.
- 7. Click each instance ID in the ID/Name of Instance at DR Site column to verify the data and applications on ECS instances at the disaster recovery site.

column, and then choose Forward Protection > Failover Completed. In the message that appears, click OK.

The protection group enters the Failover Confirmed state. In this case, the protected instances at the production site are in the Stopped state, and the ECS instances at the disaster recovery site are in the Running state.

Reverse protection: Perform a failback

1. On the Protection Group tab, find the protection group that you want to manage, move the pointer over the i i con in the Actions column, and then

choose Reverse Protection > Enable Replication.

- In the Enable Replication panel, configure and confirm resources, and click Confirm. The protection group enters the Reverse Replicating Data state.
- 3. Find the protection group, move the pointer over the i icon in the Actions column, and then choose **Reverse Protection > Failback**. In the dialog

box that appears, select **Switch Now** as Type and click **OK**.

The protection group enters the Failback in Progress state.

() Important

If you perform this operation, some data may be lost. Proceed with caution.

After the failback is complete, the protection group and its protected instances enter the Failback Completed state. In this case, the protected instances at the production site are in the Running state, and the ECS instances at the disaster recovery site are in the Stopped state.
4. After you confirm that the failback is complete, go back to the Protection Group tab. Find the protection group, move the pointer over the stopped state.

the Actions column, and then choose Reverse Protection > Failback Completed. In the message that appears, click OK.

⑦ Note After the failback is complete, the state of the protection group changes to Initializing. You can start forward replication again.

6.4. Health center

6.4.1. View current alerts

The Current Alerts page displays a list of current alerts reported by the system.

After the system reports an alert, check the operations and configurations related to Elastic Compute Service (ECS) instances at the disaster recovery site, and handle the alert accordingly. After you handle the alert or decide to ignore the alert, find the alert and click **Acknowledged** in the Actions column to clear the alert.

⑦ Note If an alert expires or disappears, you do not need to clear the alert.								
Alert Type	Alert Level	Site Pair ID	Faulty Instance	Cause	Triggered At	Actions		
vSwitch at DR Site Is Deleted	Critical	s-Ol bz4	Site Pair: s-0	secondary site network mapping(s- 0 network(vpc- n) be deleted	2023-08-06 00:20:22	Acknowledged		
VPC at DR Site Is Deleted	Critical	s-(bz4	Site Pain s-000 xz4	secondary site pair(s- (כבל)'vpc(vpc- וווט וטו אייאגטייטובא וטעלט) be deleted	2023-08-06 00:20:12	Acknowledged		

6.4.2. View the alert history

The Alert History page displays a list of historical alerts reported by the system.

You can search for alerts based on conditions such as the time when an alert was cleared and the ID of a faulty instance.

hybrid Disaster Recovery / Alert Service / Alert History									
Alert History									
	-						_		
Cleared At 🗸	2022-12-01	- 2023-12-27	8	Enter the ID of a faulty instance	Please enter the site pair ID	Select an alert type 🗸 🗸	Query		
Alert Type	Alert Level	Site Pair ID		Faulty Instance	Cause		Trig	igered At	Cleared At

6.4.3. Manage inspection tasks

Inspection tasks are used to check the availability of virtual private clouds (VPCs), networks, and security groups, and the consistency of security group rules for a site pair. If an availability or consistency exception occurs, an alert is reported.

Create an inspection task

1. Log on to the Hybrid Disaster Recovery (HDR) console.

For more information, see Log on to the HDR console

- 2. In the left-side navigation pane, choose **Alert Service > Inspection task**.
- 3. On the Inspection task page, click Create inspection task in the upper-right corner.
- 4. In the Create an inspection task dialog box, configure the parameters that are described in the following table and click OK.

Parameter	Description
Task name	The name of the inspection task. You can enter a descriptive name.
Site pair	The site pair to be inspected. Select a site pair from the Site pair drop-down list.
Inspection project	The inspection items of the inspection task. Select one or more inspection items from the Inspection project drop-down list. Valid values: VPC availability check, Network availability check, Security group availability check, and Security group policy consistency check.
Task Description	The description of the inspection task.

After the inspection task is created, it is displayed on the Inspection task page.

You can click the name of the inspection task to view the detailed configurations.

6.4.4. Manage O&M tickets

The Operation and Maintenance Work Order page displays a list of submitted O&M tickets.

You can query the processing progress of submitted O&M tickets based on conditions such as the site pair, resource ID, and ticket status.

yond Disaster Recovery / Alert Service / Operation and Maintenance Work Order									
Operation and Maintenance Work Order									
Select a site pair	Please ente	the resource ID	Please select the work order status 💙	2023-08-15	- 2023-08-16	Query		0	
Task Id	Site pair	Resource Id	Туре	Work Order Status	Creation time	Execution time	Back up data	Work order description	
oos11	c b	pg	un6 ProtectionGroupDeleteTask	Success	2023-08-16 10:43:40	2023-08-16 10:46:56	{"DrRegionId":	protection group clean, test	

After an operations administrator confirms an O&M ticket on the HDR OPS page of the Apsara Uni-manager Operations Console, the task related to the ticket starts to be run and the ticket enters the **Running** state. After the task is run, the ticket enters the **Success** state.

6.5. Cloud configuration

You can configure organizational resource mappings between cloud sites.

Background information

- Before you implement cross-cloud disaster recovery, you must contact an O&M administrator to create the peer cloud site in the Apsara Uni-manager Operations Console. Then, you can log on to the Apsara Uni-manager Management Console to view cloud sites. You can configure organizational resource mappings between cloud sites and synchronize the organizations of the local cloud site to other cloud sites to prepare for cross-cloud disaster recovery.
- Business orchestration operations such as failovers can be performed only at the primary site. Therefore, if the primary site fails, you must set the secondary site as the primary site so that you can perform failovers or recovery operations on a protection group.

Synchronize organizations of the local cloud site to other cloud sites

This operation synchronizes the organizations of the local cloud site and Hybrid Disaster Recovery (HDR) authorization status to other registered cloud sites. This allows you to create and maintain organizational resource mappings between cloud sites. The data to be synchronized includes the list of organizations and their sub-organizations that the current account can access and relevant resource sets.

- 1. Log on to the HDR console
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click Cloud configuration.
- 3. On the List of Cloud Sites page, find the cloud site whose data you want to synchronize to other cloud sites and click Synchronize on-premises cloud organizations in the Operation column.

Hybrid Disas	Hybrid Disaster Recovery / Could configuration List of Cloud Sites								
								C	
Cloud site	ID Cloud site name	regions	Proxy	Link status	Recent check time	Current cloud	Master-slave role ③	Operation	
8699		1		✓ Connected		Yes		Synchronize on-premises cloud org	
cccd	14	D1	-	✓ Connected	2023-08-16 10:45:13	No	From	SetMaster Organize resource ma	

4. In the Synchronize on-premises cloud organizations message, click OK.

A message is displayed, indicating that the synchronization is initiated.

Manage organizational resource mappings

HDR must be granted the required permissions to create organizational resource mappings between cloud sites. If HDR is not granted the permissions for a cloud site, access HDR at the cloud site to grant the permissions. Then, synchronize the organizations of the cloud site to other cloud sites and create organizational resource mappings at the primary site.

1. Log on to the HDR console.

- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click Cloud configuration.
- 3. On the List of Cloud Sites page, find the peer cloud site and click Organize resource mapping in the Operation column.
- 4. In the Organize resource mapping panel, click Add organization resource mapping.

5. In the Add organization resource mapping dialog box, configure the parameters that are described in the following table, and click OK.

	Local Configuration		Peer Configuration	
ud		4	14	
ganization	请选择	→ ≠	请选择	~
ganization AccessKey ID		4		
ganization AccessKey Secret		4		
ource Group	请选择	→ ⇒	请选择	Add

Section	Parameter	Description	
Local Configuration	Organization	The organization at the primary site.	
	AccessKey ID of Organization	The AccessKey pair of the organization at the primary site.	
	AccessKey Secret of Organization		
	Resource Group	The resource group of the organization at the primary site.	
Peer Configuration	Organization	The organization at the secondary site.	
	AccessKey ID of Organization	The Accession pair of the organization at the secondary site	
	AccessKey Secret of Organization	The Accessively pair of the organization at the secondary site.	
	Resource Group	The resource group of the organization at the secondary site.	

After the organization resource mapping is added, the mapping is displayed as Normal in the Organize resource mapping panel.

Change the role of a cloud site

Business orchestration can be performed only at the primary site. Therefore, if the primary site fails, you must set the secondary site as the primary site so that you can perform failovers or recovery operations on a protection group.

After a fault occurs, if you need to fail over or back business to an available site, but the available site is not the primary site, you must set the peer primary site as the secondary site. This way, you can manage the protection group and perform the failover or failback operation at the available site.

- 1. Log on to the HDR console.
- For more information, see Log on to the HDR console.
- 2. In the left-side navigation pane, click Cloud configuration.
- 3. On the List of Cloud Sites page, find the cloud site that you want to set as the primary site and click SetMaster in the Operation column.
- 4. In the message that appears, click **OK**.

A message is displayed, indicating that the switchover is initiated. After the switchover is complete, you can manage the protection group and perform the failover or failback operation at the primary site.

7.CDS CM 7.1. CDS Configuration Manager

Before you use Cloud Defined Storage Configuration Manager (CDS CM), you must learn about the product overview, basic concepts, features, and configuration process.

Overview

CDS CM is a cloud service that allows you to manage configurations of CDS storage in a centralized manner. CDS CM provides enterprises with onestop analysis and management of storage assets of all their Alibaba Cloud tenants and storage services. CDS CM provides the following benefits:

- Cross-tenant: provides cross-tenant data aggregation, analysis, and query. You can flexibly switch to each tenant for storage resource management.
 Cross-service: provides consistent storage management capabilities for various storage services. Besides, CDS CM can combine various storage service capabilities to meet cross-service scenarios.
- Cross-O&M: collects basic O&M data for O&M resources and provides the storage management experience of a super administrator similar to that of traditional storage vendors.

Basic concepts

Before using CDS CM, you must learn about the basic concepts of the involved resources.

Concept	Related service	Description
System tenant	CDS CM	System tenants are used for system resource management.
Data tenant	CDS CM	CDS CM uses level-1 organizations of Apsara Uni-manager as the minimum granularity for storage resource isolation. These level-1 organizations are referred to as tenants in CDS CM.
Storage set	EBS	You can classify Elastic Block Storage (EBS) clusters based on dimensions such as the business type, and associate the EBS clusters with different partitions in a storage set. This allows you to isolate EBS clusters of different business types.
Storage space	OSS	A bucket is a container that is used to store objects in Object Storage Service (OSS). All objects are stored in buckets.
Cloud disk	EBS	Cloud disks are block-level storage devices provided by Apsara Stack for Elastic Compute Service (ECS) instances. Cloud disks use the triplicate mechanism and support erasure coding (EC). Cloud disks are classified by performance into premium performance disks and standard performance disks, and are classified by usage into system disks and data disks.

Features

Feature	Operation	Description	References
Overview	Product overview	You can learn about the overall resource usage and distribution of CDS by viewing my assets, storage type analysis, and tenant resource usage distribution.	Product overview
Perceurse management	Manage tenant resources	You can manage resources such as cloud disks and buckets of different tenants, including searching for resources, editing resource information, and deleting or releasing resources.	Manage tenant resources
Resource management	View resource topology	The resource topology displays the topological relationships between the current resource and other resources. You can view the ownership of the resource.	View resource topology
Monitoring reports	Manage resource reports	Resource reports display the information and usage of all system resources, including tenants, clusters, cloud disks, buckets, and storage sets.	Manage resource reports
Intelligent analysis	Query security assessment information	You can understand the resource security status by viewing distribution of encrypted cloud disks and bucket access modes.	View security assessment information
System management	Tenants	After tenant information is collected, you can grant CDS CM the permissions to access tenant resources. CDS CM then implements centralized configuration management on tenant resources.	Manage tenants
	Task management	CDS CM executes built-in tasks or custom tasks to collect data, generate reports, and refresh tenant information. If a built-in task does not meet your requirements, create a custom task. After you create a custom task, you can execute, terminate, and delete the task.	Manage tasks

Configuration process

1. Configure organizations. For more information, see Configure organizations.

2. Initialize CDS CM. For more information, see Initialize CDS CM.

7.2. Log on to the Apsara Uni-manager Management Console

This topic describes how to log on to the Apsara Uni-manager Management Console.

Prerequisites

• The URL of the Apsara Uni-manager Management Console is obtained from the deployment personnel before you log on to the Apsara Uni-manager Management Console.

• We recommend that you use the Google Chrome browser.

Procedure

- 1. In the address bar, enter the URL of the Apsara Uni-manager Management Console. Press the Enter key.
- 2. Enter your username and password.

Obtain the username and password that you can use to log on to the console from the operations administrator.

? Note

When you log on to the Apsara Uni-manager Management Console for the first time, you must change the password of your username. Your password must meet complexity requirements. The password must be 10 to 32 characters in length and must contain at least two of the following character types:

- Uppercase or lowercase letters
- Digits
- $\circ~$ Special characters, which include ! @ # \$ %~

3. Click Log On.

4. If your account has multi-factor authentication (MFA) enabled, perform corresponding operations in the following scenarios:

• It is the first time that you log on to the console after MFA is forcibly enabled by the administrator.

- a. On the Bind Virtual MFA Device page, bind an MFA device.
- b. Enter the account and password again as in Step 2 and click ${\bf Log}~{\bf On}.$
- c. Enter a six-digit MFA verification code and click **Authenticate**.
- You have enabled MFA and bound an MFA device.

Enter a six-digit MFA authentication code and click **Authenticate**.

? Note

For more information, see the Bind a virtual MFA device to enable MFA topic in Apsara Uni-manager Operations Console User Guide.

7.3. Basic settings

7.3.1. Configure organizations

After you create a level-1 organization and grant CDS CM the permissions to view and perform operations on resources in the organization, CDS CM automatically collects the storage resources of the organization for configuration and management.

Background information

CDS CM uses level-1 organizations of Apsara Uni-manager as the minimum granularity for storage resource isolation. These level-1 organizations are referred to as tenants in CDS CM.

Step 1: Create a level-1 organization

- 1. Log on to the Apsara Uni-manager Management Console.
- 2. In the top navigation bar, click Enterprise.
- 3. In the left-side navigation pane, choose Resources > Organizations.
- 4. Select the root organization and click **Create Organization**.
- 5. In the Add Organization dialog box, specify a name for the organization and then click OK.

Step 2: Create a RAM role

You can assign the AliyunCdsCmDefaultRole service-linked role to CDS CM. This way, CDS CM is authorized to view and perform operations on resources in the organization. Perform the following steps:

- 1. Go to the Service-linked Roles page.
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, click Configurations.
- iii. In the left-side navigation pane, click Service-linked Roles.
- 2. On the Service-linked Roles page, click Create Service-linked Role.
- 3. On the page that appears, select the organization for the RAM role and set Service Name to CDS Configuration Manager.
- 4. Click **OK**.

On the Service-linked Roles page, you can view the information about the AliyunCdsCmDefaultRole service-linked role.

- The following items describe the AliyunCdsCmDefaultRole service-linked role:
- Click the Role Details tab to view the following information:
- Role Name: AliyunCdsCmDefaultRole

Trust Policy description:

- Click the Role Policy tab to view the following information:
- Policy Name: AliyunCdsCmDefaultRolePolicy
- Policy Description:



7.3.2. Initialize CDS CM

When you use CDS CM for the first time, initialize CDS CM as prompted.

Prerequisites

Organizations are configured as data tenants of CDS CM. For more information, see Configure organizations.

() Important If no organizations are configured, you can use only system tenants after CDS CM is initialized. If you want to use data tenants, you must configure organizations and grant permissions to CDS CM.

Procedure

- 1. Log on to the CDS CM console. .
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Storage > CDS Configuration**.
- 2. In the CDS CM Initialization Assistant dialog box, click Next.
- 3. CDS CM automatically collects organization information. After data is collected, click Next.

4. Select the organization that you want to use as the data tenant of CDS CM and grant permissions to CDS CM. ClickNext.

(?) Note If you do not select an organization, you can collect tenant data after CDS CM is initialized. For more information, see Manage tenants.

Dimmed organizations cannot be managed by CDS CM because the service-linked role of CDS CM does not exist in these organizations. If you require CDS CM to manage a dimmed organization, create a RAM role for CDS CM in the corresponding organization. For more information, see Create a RAM role.

 Optional:CDS CM automatically collects storage resources. After data is collected, click Finish. CDS CM periodically collects storage resources in the authorized organization. You can also create a storage resource collection task for data collection. For more information, see Manage tasks.

7.4. Product overview

You can learn about the overall resource usage and distribution of CDS by viewing my assets, storage type analysis, and tenant resource usage distribution.

Procedure

1. Log on to the CDS CM console. .

- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Storage > CDS Configuration**.
- 2. On the **Overview** page, view my assets, storage type analysis, and tenant resource usage distribution.

My assets

View the number of system tenants, the number of data tenants, the number of storage sets, the number of buckets, the number of cloud disks, and cloud disk capacity (in TB).

() Important If no organizations are configured, you can use only system tenants after Cloud Defined Storage Configuration Manager (CDS CM) is initialized. If you want to use data tenants, you must configure organizations and grant permissions to CDS CM. For more information, see Configure organizations.

Storage type analysis

View the percentage of disk storage or bucket storage.

Select Cloud Disk or Bucket from the drop-down list in the upper-right corner of the Storage type analysis section to view the related storage information of the corresponding resource.

• Tenant resource usage distribution

View the number of used resources or resource capacity (in GB) of different tenants.

Select Number of resource usages or Resource capacity from the drop-down list in the upper-right corner of the Tenant resource usage distribution section to view the resource usage of the corresponding tenant.

Click the column chart of resource usage for a tenant. On the **Tenant Resources** page, view and manage the resources of the tenant. For more information, see Manage tenant resources.

7.5. Resource management

7.5.1. Manage tenant resources

You can manage resources such as cloud disks and buckets of different tenants, including searching for resources, editing resource information, and deleting or releasing resources.

- 1. Log on to the CDS CM console. .
 - i. Log on to the Apsara Uni-manager Management Console.
 - ii. In the top navigation bar, choose **Products > Storage > CDS Configuration**.
- 2. In the left-side navigation pane, choose **Resource Management > Tenant Resources**.
- 3. On the Tenant Resources page, click the tenant name to view the basic tenant information and manage the resources owned by the tenant.
 - View tenant information

View the tenant ID, tenant name, tenant type, and role status.

- Manage cloud disks
 - On the **Cloud Disk** tab, view the cloud disk list and perform related operations.

Operation	Description
Search for cloud disks	Search for the required cloud disk by cloud disk name, UUID, properties, type, or status. You can also click Advanced Filter to search for cloud disks based on one or more conditions.
View resource topology	View the resource topology of a cloud disk, such as the tenant information, cloud disk information, and cluster of the cloud disk. Click the UUID of a cloud disk to view the resource topology of the cloud disk on th Resource Topology page. For more information, see View resource topology.
Edit cloud disk information	Edit the cloud disk name or description. a. Click Edit in the Operation column of the cloud disk. b. In the Edit Cloud Disk Editing dialog box, change the cloud disk name or description. c. Click OK .
Release cloud disks	Release the cloud disks that you no longer need. Important The cloud disks must be data disks and in thePending state. If a data disk is mounted on an instance, you must unmount the disk from the instance before you can release the disk. Click Release in the Operation column of the cloud disk. b. In the message that appears, click Release.

Manage buckets

On the **Bucket** tab, view the list of buckets and perform related operations.

Operation	Description
Search for buckets	Search for the required bucket by bucket name, read and write permissions, or storage type. You can also click Advanced Filter to search for buckets based on one or more conditions.
View resource topology	View the resource topology of a bucket, such as the tenant, bucket, location, and cluster information. Click a bucket name to view the resource topology of the bucket on the Resource Topology page. For more information, see View resource topology.

Edit the read and write permissions on buckets	 Edit the read and write permissions on a bucket. a. Click Edit in the Operation column of the bucket. b. In the Edit Bucket dialog box, specify the read and write permissions on the bucket. Valid values of the read and write permissions: Private: Only the bucket owner and authorized users have the read and write permissions on the objects in the bucket. Public Read: Only the bucket owner can perform write operations on the objects in the bucket. Other users, including anonymous users, can perform only read operations on the objects in the bucket. Public Read/Write: All users, including anonymous users, have the read and write permissions on the objects in the bucket. Fees incurred by such operations are paid by the bucket owner. Exercise caution when you configure this option. c. Click OK.
Delete a bucket	Delete a bucket that you no longer need. Important Make sure that all objects and parts in the bucket are deleted. Deleted objects, parts, and buckets cannot be recovered. Proceed with caution. a. Click Delete in the Operation column of the bucket that you want to delete. b. In the Delete message, click OK.

Manage storage sets

On the **Save set** tab, view the storage set list and perform related operations.

Operation	Description
Search for storage sets	Search for the required storage set by storage set name, UUID, region, or zone. You can also click Advanced Filter to search for storage sets based on one or more conditions.
View resource topology	View the resource topology of a storage set, such as the tenant information, storage set information, storage partitions of the storage set, and clusters bound with the storage partitions. Click the name of a storage set to view the resource topology of the storage set on th Resource Topology page. For more information, see <u>View resource</u> topology.
Edit the name of a storage set	 Edit the name of the storage set. a. Click Edit in the Operation column of the storage set. b. In the Edit Save Set dialog box, change the name of the storage set. c. Click OK.
Delete a storage set	Delete a storage set that you no longer need. ① Important Make sure that all cloud disks in the storage set are deleted. You can delete a storage set only if all cloud disks in the storage set are deleted. a. Click Delete in the Operation column of the storage set. b. In the message that appears, click OK.

7.5.2. View resource topology

The resource topology displays the topological relationships between the current resource and other resources. You can view the ownership of the resource.

Background information

Different types of resources have different related resources. The following table describes the resources.

Resource	Related resources
Cloud disk	Tenant: the tenant to which the cloud disk belongs.ECS: the ECS instance to which the cloud disk is attached.Cluster: the cluster to which the cloud disk belongs.
Bucket	Tenant: the tenant to which the bucket belongs.Location: the region where the bucket resides.Cluster: the cluster to which the bucket belongs.
Storage set	 Tenant: the tenant to which the storage set belongs. Partition: the partitions included in the storage set. You can bind a partition to or unbind the partition from a cluster. Cluster: the cluster to which a partition in a storage set is bound. Note The resource is displayed only after the partition is bound to the cluster.

1. Log on to the CDS CM console. .

i. Log on to the Apsara Uni-manager Management Console.

ii. In the top navigation bar, choose **Products > Storage > CDS Configuration**.

2. In the left-side navigation pane, choose **Resource Management > Resource Topology**.

3. On the **Resource Topology** page, enter the resource keyword and select the required resource from the matching result.

② Note Fuzzy search is supported when you query resources. You can enter a keyword to match all resources that contain the keyword. To search for a specified resource, enter the complete cloud disk UUID, bucket name, or storage set UUID.

After you select a resource, the system displays the topological relationships of the resource. This example uses storage sets.

- Click a storage resource in the resource topology to view the tenant properties and O&M properties of the storage resource.
- Click the related resources in the resource topology to view their properties. For example, you can click a tenant to view the tenant properties.
 For a storage set, double-click a storage partition. In the Edit Partition Cluster dialog box, select a cluster and click OK. You can bind the storage partition to or unbind the storage partition from the cluster.

in its original size, or click $\hfill _{\hfill }$ to show the topology in autofit mode.

7.6. Monitoring reports

7.6.1. Manage resource reports

Resource reports display the information and usage of all system resources, including tenants, clusters, cloud disks, buckets, and storage sets.

Background information

You can use the following methods to generate reports. Select a method based on your business requirements.

• Manually generate a report for one time: The report at the current time is generated for one time.

 Automatically generate reports periodically: You can create an automatic report task for the system to generate reports based on the scheduling type.

In a report, the tenant information is displayed on the **Tenant** sheet, the cluster information on the **Cluster** sheet, the cloud disk information on the **CludDisk** sheet, the bucket information on the **Bucket** sheet, and the storage set information on the **StorageSet** sheet.

Create a report

- 1. Log on to the CDS CM console. .
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Storage > CDS Configuration**.
- 2. In the left-side navigation pane, choose Monitoring Reports > Resource Reports.
- On the Resource Reports page, click Create a report. In the Create a report message, the report creation progress is displayed.
- After the report is created, you can view the generated report information and perform required operations on the Report file tab.
- Download a report to your local computer

Find the report that you want to download and click **Download** in the **Operation** column. Then, save and view the report on your local computer.

Delete a report

() Important Reports cannot be restored after they are deleted. Proceed with caution.

Find the report that you want to delete and click Delete in the Operation column. In the Delete report message, click OK.

Create an automatic report task

- 1. Log on to the CDS CM console. .
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > CDS Configuration.
- 2. In the left-side navigation pane, choose Monitoring Reports > Resource Reports.
- 3. On the Resource Reports page, click Create an automatic report task.
- 4. Configure and start the task.
 - i. In the **Create a task** dialog box, configure parameters including Task name, Scheduling type, Scheduling frequency, and Maximum number of historical tasks, and click **OK**.
 - For more information about the parameters, see Manage tasks.
- ii. On the **Task Management** page, find the task that you want to start and click **Execute** in the **Operation** column. Alternatively, move the pointer over the icon and click **Periodic execution** to start the task.
- 5. Return to the **Resource Reports** page. On the **Report Tasks** tab, view the status of the report task. After the report task is executed, you can perform operations as required.
 - Download a report to your local computer

Find the report task and click Download in the Operation column. Then, save and view the report on your local computer.

Delete a report task

() Important After a report task is deleted, the report of the task cannot be restored. Proceed with caution.

Find the report task that you want to delete and click Delete in the Operation column. In the Delete message, click OK.

7.7. Intelligent analysis

7.7.1. View security assessment information

You can understand the resource security status by viewing distribution of encrypted cloud disks and bucket access modes.

Procedure

- 1. Log on to the CDS CM console. .
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose **Products > Storage > CDS Configuration**.
- 2. In the left-side navigation pane, choose Intelligent Analysis > Security Assessment.

3. On the Security Assessment page, view distribution of encrypted cloud disks and bucket access modes.

Once If the access control mode of the bucket is Unknown, CDS CM cannot obtain the bucket access control mode due to a bucket exception.

7.8. System management

7.8.1. Manage tenants

After tenant information is collected, you can grant CDS CM the permissions to access tenant resources. CDS CM then implements centralized configuration management on tenant resources.

Background information

CDS CM uses level-1 organizations of Apsara Uni-manager as the minimum granularity for storage resource isolation. These level-1 organizations are referred to as tenants in CDS CM.

Precautions

Dimmed organizations cannot be managed by CDS CM because the service-linked role of CDS CM does not exist in these organizations. If you require CDS CM to manage a dimmed organization, create a RAM role for CDS CM in the corresponding organization. For more information, see Create a RAM role.

Procedure

- 1. Log on to the CDS CM console. .
 - i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose $\mbox{Products}$ > $\mbox{Storage}$ > CDS Configuration.
- 2. In the left-side navigation pane, choose System Management > Tenants.
- By default, CDS CM automatically collects tenant resources. You can also click Collect Tenant to collect the latest tenant information.
- 3. On the **Tenants** page, select the required tenant and click **Submit**.

② Note To manage a dimmed organization, click RAM service role in the lower part of the page. In the Authorize Access to Cloud Resources dialog box, select the required organization and click Approve to grant CDS CM the permissions to access the organization.

4. In the Storage Resource Collection dialog box, CDS CM collects the storage resources of the organization. After data is collected, click Complete.

7.8.2. Manage tasks

CDS CM executes built-in tasks or custom tasks to collect data, generate reports, and refresh tenant information. If a built-in task does not meet your requirements, create a custom task. After you create a custom task, you can execute, terminate, and delete the task.

Background information

CDS CM provides built-in tasks and custom tasks.

- Built-in tasks include ManualDataCollectJob (manual data collection task), ManualReportJob (manual report task), AutoDataCollectJob (automatic data collection task), and ManualRefreshTenantsJob (manual tenant information refreshing task). Built-in tasks cannot be manually executed, terminated, or deleted.
- Custom tasks include DataCollectJob (data collection task), ReportJob (report task), and RefreshTenantsJob (tenant information refreshing task). Custom tasks can be manually executed, terminated, and deleted.

Procedure

- 1. Log on to the CDS CM console. .
- i. Log on to the Apsara Uni-manager Management Console.
- ii. In the top navigation bar, choose Products > Storage > CDS Configuration.
- 2. In the left-side navigation pane, choose **System Management > Task Management**.
- 3. Create a task.
 - i. On the Task Management page, click Create a task.

ii. In the Create a task dialog box, configure the task parameters. The following table describes the parameters.

Parameter	Description
Task name	The task name, which uniquely identifies a task.
Template	 The task template provided by the system, which allows you to quickly create a task of the corresponding type. Valid values: DataCollectJob: collects information about storage resources of a tenant. ReportJob: generates resource reports. RefreshTenantSJob: refreshes tenant information.
Task type	 The task execution method. Valid values: One-time execution: The task is executed for one time. Periodic scheduling: The task is executed periodically based on the specified scheduling type and scheduling frequency.
Scheduling type	 The periodic scheduling mode. Valid values: Fixed Frequency Scheduling: After the previous task is completed, the next task is started at a fixed interval. Fixed Interval scheduling: After the previous task is started, the next task is started at a fixed interval. If the previous task is not completed at the current time, the next task is started immediately after the previous task is completed. This option is available only if you setTask type to Periodic scheduling.
Scheduling frequency	The scheduling interval. Unit: seconds. The value must be a positive integer. This option is available only if you set Task type to Periodic scheduling .
Maximum number of historical tasks	The maximum number of historical tasks that can be retained by the system.
Timeout	The timeout period of task execution. Unit: seconds. If the execution time of a task exceeds the timeout period, the task fails to be executed.
Description	The task description.

iii. Click **OK**.

4. Execute a task.

- For a one-time task, click **Execute** in the **Operation** column of the custom task.
- For a periodic task, click Execute in the Operation column of a custom task to execute the task for one time, or move the pointer over the ••••

icon and click **Periodic execution** to execute the task periodically.

Related operations

After a task is created, perform the required operations on the Task Management page.

Operation	Description
Search for a task	Search for the required task by task name, template type, scheduling status, task type, or scheduling type. You can also click Advanced Filter to search for tasks based on one or more conditions.
Execute a task	Execute a task for one time. Click Execute in the Operation column of a custom task.
Execute a task periodically	Execute a task periodically. To execute a periodic custom task, move the pointer over the •••• icon and click Periodic execution .
Terminate a task	Terminate a custom task. Move the pointer over the •••• icon and click Termination . In the Termination message, click OK .
Delete a task	Delete a custom task that you no longer need. Move the pointer over the •••• icon and click Delete . In the Delete message, click OK .

Manage historical execution information of tasks.	 View or delete the historical execution information of a task. Click the task name. In the Task Instance panel, perform operations based on your business requirements. View historical task execution information View the historical execution information of a task, including the start time, end time, elapsed time, and execution result.
	Delete historical task execution information
	Find a task instance that you want to delete and click Delete in the Operation column. In the Delete message, click OK .