

Alibaba Cloud

Apsara Stack Enterprise

Cloud Defined Storage User Guide

Product Version: v3.16.2

Document Version: 20220916

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.

1. 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. The names owned by 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.
 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.
 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 .
<code>Courier font</code>	Courier font is used for commands	Run the <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

1. Overview	21
2. OSS	22
2.1. What is OSS?	22
2.1.1. Features	24
2.1.1.1. Manage buckets	24
2.1.1.1.1. Create a bucket	24
2.1.1.1.2. ACL	24
2.1.1.1.3. Static website hosting	25
2.1.1.1.4. Logging	26
2.1.1.1.5. Lifecycle rules	26
2.1.1.2. Manage objects	28
2.1.1.2.1. Upload objects	28
2.1.1.2.2. ACL	28
2.1.1.2.3. Download objects	28
2.1.1.2.4. Search for objects	29
2.1.1.2.5. Manage objects by using directories	29
2.1.1.2.6. Object tagging	31
2.1.1.3. Data security	33
2.1.1.3.1. Erasure coding	33
2.1.1.3.2. Resource isolation	34
2.1.1.3.3. Disaster recovery	34
2.1.1.3.4. Access permissions and account authorization	34
2.1.1.3.5. Server-side encryption	35
2.1.1.3.6. Client-side encryption	37
2.1.1.3.7. Versioning	39
2.1.1.4. Data processing	43

2.1.1.4.1. IMG	43
2.1.1.4.2. Video snapshots	43
2.2. Usage notes	44
2.3. Quick start	45
2.3.1. Log on to the OSS console	45
2.3.2. Create a bucket	46
2.3.3. Upload objects	49
2.3.4. Obtain object URLs	50
2.4. Buckets	51
2.4.1. View bucket information	51
2.4.2. Delete buckets	51
2.4.3. Modify bucket ACLs	51
2.4.4. Configure static website hosting	52
2.4.5. Configure hotlink protection for a bucket	53
2.4.6. Configure CORS	54
2.4.7. Configure lifecycle rules	55
2.4.8. Configure storage quota	57
2.4.9. Configure bucket tagging	58
2.4.10. Configure zone-disaster recovery	58
2.4.11. Versioning	60
2.4.12. Configure server-side encryption	61
2.4.13. Bind a VPC	62
2.4.14. Configure CRR	62
2.4.15. Configure cross-cloud replication	64
2.4.16. Log management	66
2.4.16.1. Configure logging	66
2.4.16.2. Real-time log query	66
2.4.17. Image processing	69

2.4.17.1. Configure image styles	69
2.4.17.2. Configure source image protection	70
2.4.18. Grants permissions to a role	71
2.5. Objects	74
2.5.1. Search for objects	74
2.5.2. Configure object ACLs	75
2.5.3. Configure object metadata	75
2.5.4. Create directories	77
2.5.5. Download objects	78
2.5.6. Delete objects	78
2.5.7. Manage parts	79
2.5.8. Configure object tagging	79
2.5.9. Configure bucket policies to authorize other users to a...	80
2.6. Add OSS paths	85
2.7. Create single tunnels	85
2.8. CSG	86
2.8.1. What is CSG?	86
2.8.2. Usage notes	87
2.8.3. Quick start	88
2.8.4. File gateways	93
2.8.4.1. Manage file gateways	93
2.8.4.2. Manage shares	94
2.8.4.3. Manage cache disks	100
2.8.4.4. Create an SMB user	100
2.8.4.5. Access shares	101
2.8.4.5.1. Access an NFS share	101
2.8.4.5.2. Access an SMB share	103
2.8.4.6. Update a gateway	104

2.8.4.7. Appendix	104
2.8.4.7.1. Permissions required by a gateway to operate O...	104
3.NAS	106
3.1. What is NAS?	106
3.2. Usage notes	106
3.3. Quick start	108
3.3.1. Log on to the NAS console	108
3.3.2. Create a file system	108
3.3.3. Create a permission group and add rules	109
3.3.4. Create a mount target	111
3.3.5. Mount an NFS file system	113
3.3.6. Mount an SMB file system	116
3.4. File systems	118
3.4.1. View the details of a file system	118
3.4.2. Delete a file system	119
3.4.3. Scale up a file system	119
3.5. Mount targets	119
3.5.1. View mount targets	119
3.5.2. Enable or disable a mount target	120
3.5.3. Delete a mount target	120
3.5.4. Modify the permission group of a mount target	121
3.6. Permission groups	121
3.6.1. View permission groups	121
3.6.2. Delete a permission group	122
3.6.3. Manage permission group rules	122
3.7. Directory-level ACLs that grant the read and write access	123
3.7.1. Overview	123
3.7.2. Features	124

3.7.3. Use POSIX ACLs to control access	134
3.7.4. Use NFSv4 ACLs to control access	138
4.EBS	142
4.1. EBS	142
4.1.1. Features	144
4.1.1.1. Overview	144
4.1.1.2. Triplicate storage	145
4.1.1.3. Erasure coding	146
4.1.1.4. EBS device encryption	146
4.1.1.5. EBS device resizing	147
4.1.1.6. Snapshots	148
4.1.1.6.1. Overview	148
4.1.1.6.2. Mechanisms	148
4.1.1.6.3. Specifications of ECS Snapshot 2.0	149
4.1.1.7. Async replication	150
4.2. Log on to the Apsara Uni-manager Management Console	154
4.3. Getting started	155
4.3.1. Create a disk	155
4.3.2. Attach disks	157
4.3.3. Partition and format a disk	159
4.3.3.1. Format a data disk on a Linux instance	159
4.3.3.2. Format a data disk on a Windows ECS instance	162
4.4. Disks	162
4.4.1. Search for and view a disk	162
4.4.2. Modify the properties of a disk	163
4.4.3. Modify the name and description of a disk	164
4.4.4. Resize a disk	164
4.4.5. Enable the multi-attach feature for disks that support ...	165

4.4.5.1. Overview of disks that support NVMe	166
4.4.5.2. Enable the multi-attach feature for disks	166
4.4.6. Encrypt a disk	170
4.4.6.1. Encrypt a system disk	170
4.4.6.2. Encrypt a data disk	172
4.4.7. Roll back a disk by using a snapshot	172
4.4.8. Re-initialize disks	173
4.4.8.1. Re-initialize a system disk	173
4.4.8.2. Re-initialize a data disk	174
4.4.9. Detach a data disk	175
4.4.10. Release a data disk	178
4.5. Snapshots	179
4.5.1. Disk snapshots	179
4.5.1.1. Create a snapshot	179
4.5.1.2. Search for and view a snapshot	180
4.5.1.3. Roll back a disk by using a snapshot	181
4.5.1.4. Create a custom image from a snapshot	181
4.5.1.5. Delete one or more snapshots at a time	182
4.5.2. Snapshot-consistent groups	183
4.5.2.1. Create a snapshot-consistent group	183
4.5.2.2. Search for and view a snapshot-consistent group	184
4.5.2.3. Roll back disks by using a snapshot-consistent gro...	185
4.5.2.4. Delete a snapshot-consistent group	186
4.6. Automatic snapshot policies	186
4.6.1. Create an automatic snapshot policy	186
4.6.2. Search for and view an automatic snapshot policy	188
4.6.3. Modify an automatic snapshot policy	189
4.6.4. Enable or disable an automatic snapshot policy	189

4.6.5. Delete an automatic snapshot policy	190
4.7. Async replication	190
4.7.1. Create a replication pair	190
4.7.2. View replication pairs	192
4.7.3. Modify a replication pair	192
4.7.4. Activate a replication pair to enable async replication	193
4.7.5. Stop a replication pair to disable async replication	194
4.7.6. Use the async replication feature to implement disaster... ..	195
4.7.7. Delete a replication pair	196
4.8. Storage sets	197
4.8.1. Create a storage set	197
4.8.2. View storage sets	198
4.8.3. Modify a storage set	198
4.8.4. Delete a storage set	199
4.9. Replication pair-consistent groups	199
4.9.1. Create a replication pair-consistent group	199
4.9.2. View replication pair-consistent groups	200
4.9.3. Modify a replication pair-consistent group	201
4.9.4. Add replication pairs to a replication pair-consistent gr... ..	201
4.9.5. Remove replication pairs from a replication pair-consist... ..	202
4.9.6. Activate a replication pair-consistent group to enable a... ..	203
4.9.7. Stop a replication pair-consistent group to disable asyn... ..	204
4.9.8. Use replication pair-consistent groups to implement dis... ..	204
4.9.9. Delete a replication pair-consistent group	206
4.10. Appendix	206
4.10.1. Manage tags	206
5. Log Service	209
5.1. What is Log Service?	209

5.2. Quick start	209
5.2.1. Procedure	209
5.2.2. Log on to the Log Service console	211
5.2.3. Obtain an AccessKey pair	212
5.2.4. Manage a project	213
5.2.5. Manage a Metricstore	214
5.2.6. Manage Logstores	216
5.2.7. Manage shards	218
5.2.8. Terms	220
5.2.8.1. Terms	220
5.2.8.2. Log	222
5.2.8.3. Log group	224
5.2.8.4. Project	224
5.2.8.5. Logstore	224
5.2.8.6. Metricstore	224
5.2.8.7. Metric	225
5.2.8.8. Shard	225
5.2.8.9. Topic	227
5.3. Data collection	228
5.3.1. Collection by Logtail	228
5.3.1.1. Overview	228
5.3.1.1.1. Logtail overview	228
5.3.1.1.2. Log collection process of Logtail	231
5.3.1.1.3. Logtail configuration files and record files	233
5.3.1.2. Installation	241
5.3.1.2.1. Install Logtail on a Linux server	241
5.3.1.2.2. Install Logtail in Windows	243
5.3.1.2.3. Set the startup parameters of Logtail	245

5.3.1.3. Logtail machine group	250
5.3.1.3.1. Overview	250
5.3.1.3.2. Configure a user identifier	251
5.3.1.3.3. Create an IP address-based machine group	252
5.3.1.3.4. Create a custom ID-based machine group	254
5.3.1.3.5. View server groups	257
5.3.1.3.6. Modify a server group	257
5.3.1.3.7. View the status of a server group	258
5.3.1.3.8. Delete a machine group	258
5.3.1.3.9. Manage a Logtail configuration	258
5.3.1.4. Collect text logs	259
5.3.1.4.1. Configure text log collection	259
5.3.1.4.2. Collect logs in simple mode	264
5.3.1.4.3. Collect logs in full regex mode	268
5.3.1.4.4. Collect logs in delimiter mode	273
5.3.1.4.5. Collect logs in JSON mode	278
5.3.1.4.6. Collect logs in NGINX mode	282
5.3.1.4.7. Collect logs in IIS mode	287
5.3.1.4.8. Collect logs in Apache mode	293
5.3.1.4.9. Configure parsing scripts	299
5.3.1.4.10. Time formats	301
5.3.1.4.11. Import historical log files	304
5.3.1.4.12. Log topics	306
5.3.1.5. Collect container logs	307
5.3.1.5.1. Overview	307
5.3.1.5.2. Install the Logtail component	308
5.3.1.5.3. Use the Log Service console to collect container... ..	313
5.3.1.5.4. Use the Log Service console to collect container... ..	324

5.3.1.5.5. Use CRDs to collect container logs in DaemonSe...	341
5.3.1.5.6. Use CRDs to collect container text logs in Sidec...	353
5.3.1.5.7. Use the Log Service console to collect container ...	367
5.3.1.5.8. Collect logs from standard Docker containers	372
5.3.1.5.9. Collect Kubernetes events	375
5.3.1.5.10. Collect container text logs	378
5.3.1.5.11. Collect container stdout and stderr logs	384
5.3.1.5.12. Collect standard Docker logs	396
5.3.1.6. Custom plug-ins	400
5.3.1.6.1. Collect MySQL binary logs	401
5.3.1.6.2. Collect MySQL query results	411
5.3.1.6.3. Collect syslogs	416
5.3.1.6.4. Customize Logtail plug-ins to process data	421
5.3.1.7. Limits	444
5.3.2. Other collection methods	447
5.3.2.1. Use the web tracking feature to collect logs	447
5.3.2.2. Collect logs over the syslog protocol	450
5.3.2.3. Use SDKs to collect logs	454
5.3.2.3.1. Producer Library	454
5.3.2.3.2. Log4j Appender	454
5.3.2.3.3. Logback Appender	454
5.3.2.3.4. Golang Producer Library	455
5.3.2.3.5. Python logging	455
5.3.2.4. Collect common logs	458
5.3.2.4.1. Collect Log4j logs	458
5.3.2.4.2. Collect Python logs	460
5.3.2.4.3. Collect Node.js logs	464
5.3.2.4.4. Collect WordPress logs	465

5.3.2.4.5. ThinkPHP logs	466
5.3.2.4.6. Collect Unity3D logs	467
5.4. Query and analysis	469
5.4.1. Log search overview	469
5.4.2. Log analysis overview	470
5.4.3. Configure indexes	472
5.4.4. Query and analyze logs	475
5.4.5. Download logs	477
5.4.6. Index data type	477
5.4.6.1. Overview	477
5.4.6.2. Text type	478
5.4.6.3. Numeric type	479
5.4.6.4. JSON type	480
5.4.7. Query syntax and functions	483
5.4.7.1. Search syntax	483
5.4.7.2. LiveTail	490
5.4.7.3. LogReduce	491
5.4.7.4. Contextual query	494
5.4.7.5. Saved search	496
5.4.7.6. Quick analysis	498
5.4.7.7. Other features	500
5.4.8. SQL syntax and functions	501
5.4.8.1. General aggregate functions	501
5.4.8.2. Security check functions	504
5.4.8.3. Map functions and operators	506
5.4.8.4. Approximate functions	516
5.4.8.5. Mathematical statistics functions	516
5.4.8.6. Mathematical calculation functions	518

5.4.8.7. String functions	520
5.4.8.8. Date and time functions	523
5.4.8.9. URL functions	529
5.4.8.10. Regular expression functions	530
5.4.8.11. JSON functions	531
5.4.8.12. Type conversion functions	534
5.4.8.13. IP functions	535
5.4.8.14. GROUP BY clause	536
5.4.8.15. Window functions	538
5.4.8.16. HAVING clause	540
5.4.8.17. ORDER BY clause	540
5.4.8.18. LIMIT syntax	541
5.4.8.19. Conditional expressions	541
5.4.8.20. Nested subquery	544
5.4.8.21. Array functions and operators	545
5.4.8.22. Binary string functions	563
5.4.8.23. Bitwise functions	564
5.4.8.24. Interval-valued comparison and periodicity-valued ...-----	564
5.4.8.25. Comparison functions and operators	568
5.4.8.26. Lambda expressions	570
5.4.8.27. Logical functions	572
5.4.8.28. Column aliases	573
5.4.8.29. JOIN queries on a Logstore and a MySQL databas...-----	574
5.4.8.30. Geospatial functions	575
5.4.8.31. Geography functions	578
5.4.8.32. JOIN clause	579
5.4.8.33. UNNEST clause	580
5.4.9. Machine learning syntax and functions	582

5.4.9.1. Overview	582
5.4.9.2. Smooth functions	584
5.4.9.3. Multi-period estimation functions	587
5.4.9.4. Change point detection functions	590
5.4.9.5. Maximum value detection function	592
5.4.9.6. Prediction and anomaly detection functions	593
5.4.9.7. Time series decomposition function	599
5.4.9.8. Time series clustering functions	600
5.4.9.9. Frequent pattern statistics function	605
5.4.9.10. Differential pattern statistics function	606
5.4.9.11. Root cause analysis function	607
5.4.9.12. Correlation analysis functions	610
5.4.9.13. Kernel density estimation function	613
5.4.10. Advanced analysis	614
5.4.10.1. Optimize queries	614
5.4.10.2. Use cases	615
5.4.10.3. Examples of time field conversion	618
5.4.11. Associate Log Service with external data sources	619
5.4.11.1. Overview	619
5.4.11.2. Associate Log Service with a MySQL database	620
5.4.11.3. Associate Log Service with an OSS bucket	622
5.4.11.4. Associate Log Service with a hosted CSV file	624
5.4.12. Visual analysis	627
5.4.12.1. Charts	627
5.4.12.1.1. Chart overview	627
5.4.12.1.2. Display query results in a table	628
5.4.12.1.3. Display query results on a line chart	629
5.4.12.1.4. Display query results on a column chart	631

5.4.12.1.5. Display query results on a bar chart	632
5.4.12.1.6. Display query results on a pie chart	633
5.4.12.1.7. Display query results on an area chart	636
5.4.12.1.8. Display query results on a single value chart	637
5.4.12.1.9. Display query results on a progress bar	641
5.4.12.1.10. Display query results on a map	643
5.4.12.1.11. Display query results on a flow chart	643
5.4.12.1.12. Display query results in a Sankey diagram	645
5.4.12.1.13. Display query results on a word cloud	646
5.4.12.1.14. Display query results on a treemap chart	647
5.4.12.2. Dashboard	648
5.4.12.2.1. Overview	648
5.4.12.2.2. Create and delete a dashboard	649
5.4.12.2.3. Manage a dashboard in display mode	649
5.4.12.2.4. Manage a dashboard in edit mode	651
5.4.12.2.5. Configure a drill-down event	653
5.4.12.2.6. Add a filter	660
5.4.12.2.7. Manage a Markdown chart	661
5.4.13. Visualized data analytics	665
5.5. Alerts	665
5.5.1. Overview	665
5.5.2. Configure an alarm	666
5.5.2.1. Configure an alert rule	666
5.5.2.2. Authorize a RAM user to manage alert rules	668
5.5.2.3. Configure alert notification methods	669
5.5.3. Modify and view an alarm	673
5.5.3.1. Modify an alert rule	673
5.5.3.2. View alert statistics	674

5.5.3.3. Manage alerts	675
5.5.4. Relevant syntax and fields for reference	676
5.5.4.1. Syntax of conditional expressions in alert rules	676
5.5.4.2. Fields in alert logs	680
5.5.5. FAQ	682
5.5.5.1. A DingTalk alert notification fails to be sent and th... ..	682
5.6. Real-time consumption	683
5.6.1. Overview	683
5.6.2. Consume log data	684
5.6.3. Consumption by consumer groups	686
5.6.3.1. Use consumer groups to consume log data	686
5.6.3.2. View the status of a consumer group	692
5.6.4. Use Storm to consume log data	693
5.6.5. Use Flume to consume log data	697
5.6.6. Use open source Flink to consume log data	700
5.6.7. Use Logstash to consume log data	706
5.6.8. Use Spark Streaming to consume log data	708
5.6.9. Use Realtime Compute to consume log data	712
5.7. Data shipping	715
5.7.1. Ship logs to OSS	715
5.7.1.1. Overview	715
5.7.1.2. Ship log data from Log Service to OSS	716
5.7.1.3. Obtain the ARN of a RAM role	720
5.7.1.4. Storage Formats	721
5.7.1.5. Decompress Snappy compressed files	723
5.8. Time series storage	725
5.8.1. Data import	725
5.8.1.1. Collect metric data from hosts	725

5.8.1.2. Import metrics collected by Telegraf	730
5.8.1.2.1. Telegraf overview	730
5.8.1.2.2. Collect metric data from MySQL servers	731
5.8.1.2.3. Collect metric data from Java applications or To... ..	732
5.8.1.2.4. Collect metric data from NGINX servers	735
5.8.1.3. Collect metric data from Prometheus	737
5.8.1.3.1. Collect metric data from Prometheus by using th... ..	738
5.8.1.3.2. Collect metric data from Prometheus by using a... ..	740
5.8.2. Query and analysis	742
5.8.2.1. Overview of query and analysis of time series data	742
5.8.2.2. Query and analyze time series data	745
5.8.3. Visualization	747
5.8.3.1. Configure a time series chart	747
5.8.3.2. Send time series data from Log Service to Grafana	747
5.9. RAM	749
5.9.1. Overview	749
5.9.2. Create a RAM role	750
5.9.3. Create a user	750
5.9.4. Create a user group	751
5.9.5. Add a user to a user group	751
5.9.6. Create a policy	752
5.9.7. Grant a RAM user the permissions to manage a project	753
5.9.8. Grant permissions to a RAM role	754
5.9.9. Use custom policies to grant permissions to a RAM use... ..	754
5.10. Monitor Log Service	759
5.10.1. Overview	759
5.10.2. Manage service logs	760
5.10.3. Log types	763

5.10.4. Service log dashboards	773
5.11. FAQ	773
5.11.1. Log collection	773
5.11.1.1. How do I troubleshoot errors that occur when I us... ..	774
5.11.1.2. What can I do if Log Service does not receive hea... ..	774
5.11.1.3. How do I query the status of local log collection?	776
5.11.1.4. How do I debug a regular expression?	788
5.11.1.5. How do I optimize regular expressions?	790
5.11.1.6. How do I use the full regex mode to collect log en... ..	791
5.11.1.7. How do I specify time formats for logs?	791
5.11.1.8. How do I configure non-printable characters in a s... ..	792
5.11.1.9. How do I troubleshoot errors that occur when I co... ..	793
5.11.1.10. How do I obtain the labels and environment varia... ..	796
5.11.2. Log search and analysis	798
5.11.2.1. FAQ about log query	798
5.11.2.2. What can I do if I cannot obtain the required res... ..	799
5.11.2.3. What are the differences between log consumption... ..	800
5.11.2.4. How do I resolve common errors that occur when	801
5.11.2.5. Why data queries are inaccurate?	803
5.11.2.6. How do I configure indexes for historical log data?	804
5.11.3. Alarm	804
5.11.3.1. FAQ about alerts	804
5.11.4. What do I do if the Forbidden.SLS::ListProject error o... ..	805

1. Overview

Cloud Defined Storage (CDS) is a distributed file storage system provided by Alibaba Cloud. CDS features high security, high reliability, low cost, centralized management, and elastic resource scaling, which allows efficient data storage and retrieval.

Based on Object Storage Service (OSS), Log Service, Apsara File Storage NAS, and Elastic Block Storage (EBS), 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.

Limit	Description
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.

ACL	Description
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.  Warning All Internet users can access objects in the bucket. This may result in unexpected access to the data in your bucket and out-of-control costs. Exercise caution when you set your bucket ACL to Public Read.
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* 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.



Notice 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-current 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.

2.1.1.2. 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.

Folder structure	Specified prefix	Search result
TestBucket	Example	Examplesrcfolder1
└─ Examplesrcfolder1		Exampledestfolder.png
├─ test.txt		
├─ abc.jpg		
└─ Exampledestfolder.png		
└─ example.txt		

- **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.

Folder structure	Specified prefix	Search result
Examplesrcfolder1	Project	Projectfolder
└─ Projectfolder		ProjectA.jpg
├─ a.txt		ProjectB.doc
├─ b.txt		
└─ ProjectA.jpg		
└─ ProjectB.doc		
└─ projectC.doc		

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`:

```
examplebucket
├── log/
│   ├── date1.txt
│   ├── date2.txt
│   └── date3.txt
└── destfolder/
    └── 2021/
        └── photo.jpg
```

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:

```
<LifecycleConfiguration>
  <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>
      <Days>30</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>r2</ID>
    <Prefix>rule2</Prefix>
    <Tag><Key>xx</Key><Value>1</Value></Tag>
    <Status>Enabled</Status>
    <Transition>
      <Days>60</Days>
      <StorageClass>Archive</StorageClass>
    </Transition>
  </Rule>
</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:

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": "oss:PutObjectTagging",
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "oss:RequestObjectTag/allow": [
            "yes"
          ]
        }
      }
    }
  ]
}
```



Notice 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:

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "oss:GetObject",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "oss:ExistingObjectTag/allow": [
            "yes"
          ]
        }
      }
    }
  ]
}
```

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:

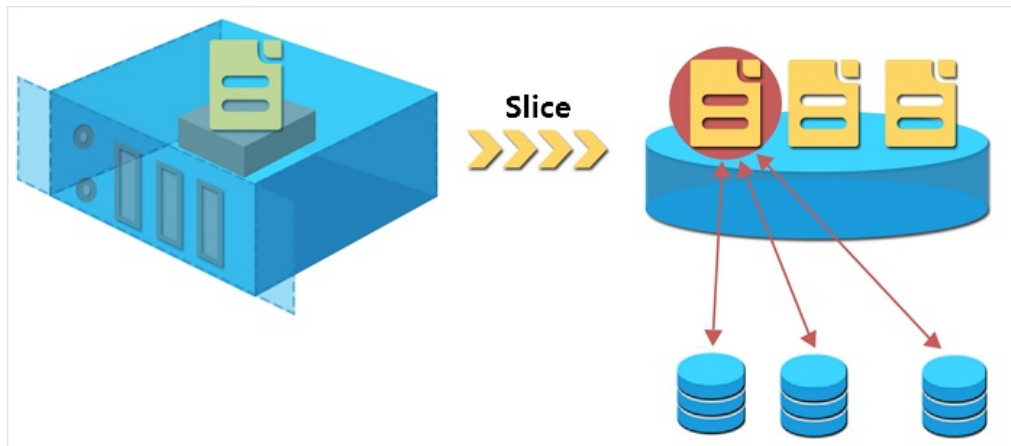
- Data fragments (m): Data is divided into m data fragments.
- Parity fragments (n): n parity fragments are computed based on the m data fragments.

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	$\frac{m}{m+n}$. 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. 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.

2.1.1.3.3. Disaster recovery

OSS provides multiple disaster recovery types to ensure data security and improve availability.

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

Type	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 when a single zone fails.
CRR	Cross-region replication (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.
Cross-cloud replication	You can use cross-cloud replication to replicate data from one cloud to another cloud. This way, you can back up data across clouds. If a cloud fails, you can switch over your business to another cloud to ensure business continuity.
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.

2.1.1.3.4. 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: <i>Public Read/Write</i> , <i>Public-Read</i> , and <i>Private</i> .
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: <i>Inherited from bucket</i> , <i>Public Read/Write</i> , <i>Public-Read</i> , and <i>Private</i> .
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.

2.1.1.3.5. 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.

Only one server-side encryption method can be used for an object at a time. OSS provides the following server-side encryption methods that you can use in different scenarios:

- Server-side encryption by using Key Management Service (SSE-KMS)

You can use the default customer master key (CMK) managed by KMS or specify a CMK to encrypt or decrypt data. This method is cost-effective because you do not need to send the data to the KMS server over networks for encryption or decryption.

Notice

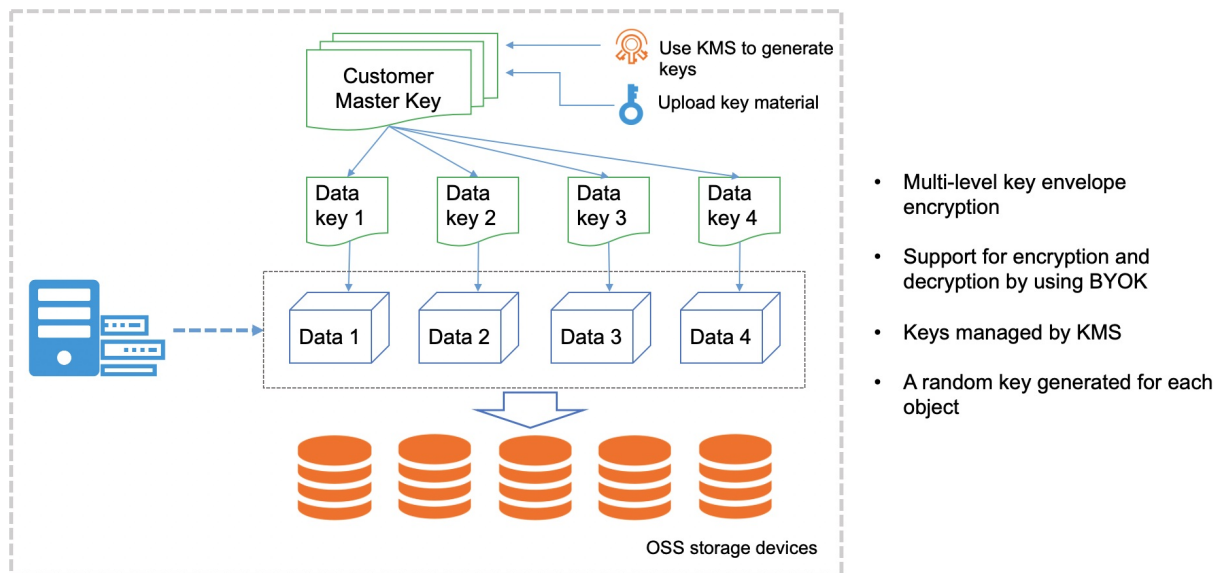
- The key used to encrypt the object is also encrypted and written into the metadata of the object.
- Server-side encryption that uses the default CMK (SSE-KMS) only encrypts the data in the object. The metadata of the object is not encrypted.

• Server-side encryption by using OSS-managed keys (SSE-OSS)

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 CMKs stored in KMS (SSE-KMS)

You can use a CMK stored in KMS to generate a data key to encrypt data. The envelope encryption mechanism further prevents unauthorized data access. KMS eliminates the need to manually maintain the security, integrity, and availability of your keys. You need only to focus on data encryption, data decryption, and digital signature generation and verification based on your business requirements.



When you use SSE-KMS to encrypt data, you can use the following keys:

• Use CMKs stored in KMS

In this method, OSS generates different data keys by using the default CMK stored in KMS to encrypt different objects, and automatically decrypts an object when the object is downloaded. OSS creates a CMK in KMS the first time you use SSE-KMS.

Use the following configuration methods:

- Configure the default server-side encryption method for a bucket

Set the default server-side encryption method to KMS for a bucket as the encryption algorithm, but do not specify a CMK ID. This way, objects uploaded to this bucket are encrypted.

- Configure an encryption method for a specified 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 `KMS`. This way, OSS uses the default CMK stored in KMS and uses the AES-256 encryption algorithm to encrypt the object.

• Use Bring Your Own Key (BYOK)

After you use the BYOK material in the KMS console to generate a CMK, OSS uses the CMK to generate different data keys to encrypt different objects. The CMK ID is recorded in the metadata of the encrypted object. Then, the objects are decrypted only when they are downloaded by users who have the permissions to decrypt the objects.

You can import your BYOK material into KMS as the CMK:

- BYOK material provided by Alibaba Cloud: When you create a key on KMS, you can select **Alibaba Cloud KMS** as the source of the key material.
- BYOK material provided by the user: When you create a key on KMS, you can select **External** for Key Material Source to import external key material.

Use the following configuration methods:

- Configure the default server-side encryption method for a bucket

Set the default encryption method to KMS as the encryption algorithm. In addition, specify the CMK ID. This way, 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 `KMS`. In addition, include the `x-oss-server-side-encryption-key-id` parameter in the request and set the parameter value to the specified CMK ID. This way, OSS uses the specified CMK stored in KMS and the AES-256 encryption algorithm to encrypt the object.

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-256` or `SM4`. This way, the object is encrypted by using SSE-OSS.

2.1.1.3.6. 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.

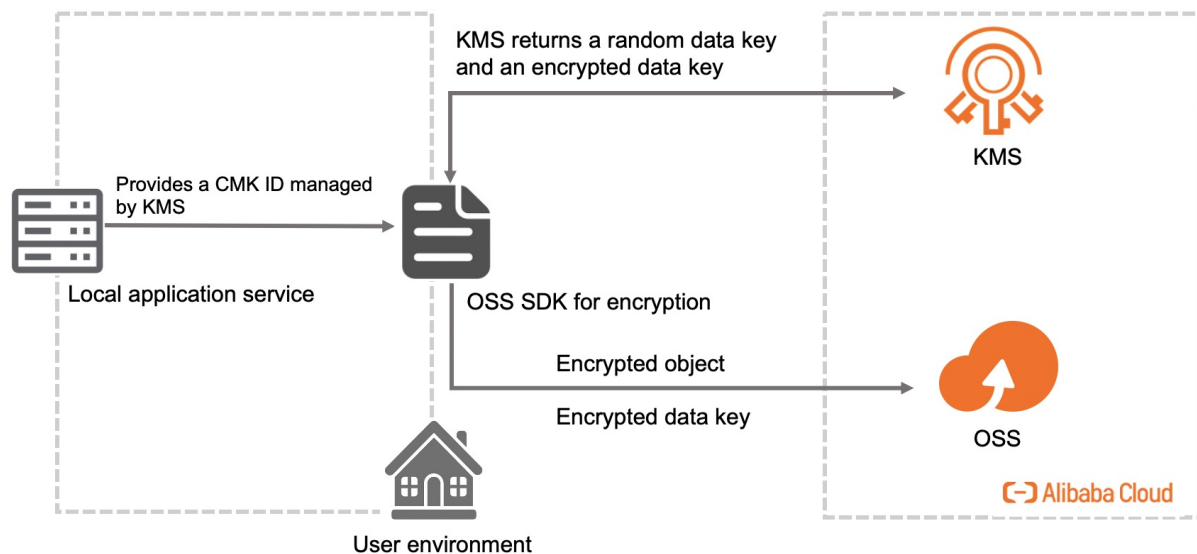
Encryption

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.

OSS supports Key Management Service (KMS)-managed CMKs and customer-managed CMKs.

Use KMS-managed CMKs

If you use KMS-managed CMKs for client-side encryption, you need only to specify the CMK ID when you upload objects instead of providing the client with a data key. The following figure shows the specific encryption process.



- Encrypt and upload an object

- i. Obtain a data key.

The client uses a specified CMK ID to request a data key used to encrypt the object from KMS. KMS returns a random data key and an encrypted data key.

- ii. Encrypt the object and upload the object to OSS.

The client uses the returned data key to encrypt the object and uploads the encrypted object and encrypted data key to OSS.

- Download and decrypt an object

- i. Download an object.

The client downloads an encrypted object and the encrypted data key of the object. The encrypted data key is included in the metadata of the object.

- ii. Decrypt the object.

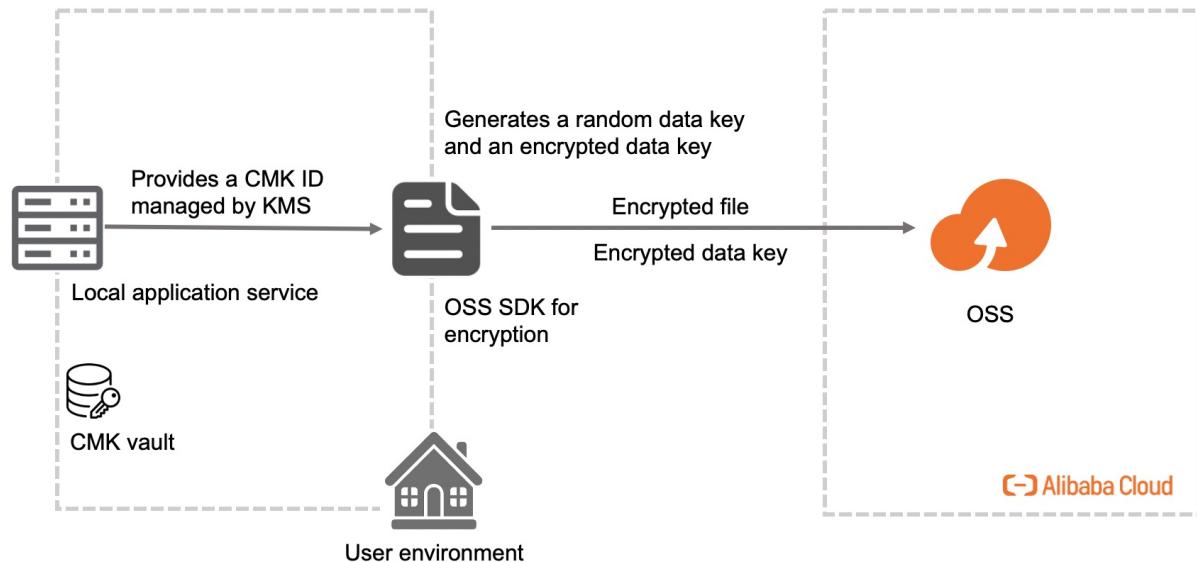
The client sends the encrypted data key and the corresponding CMK ID to KMS. KMS uses the CMK sent by the client to decrypt the encrypted data key and returns the data key to the client.

Note

- The client obtains a unique data key for each object to upload.
- To ensure data security, we recommend that you regularly rotate or update the CMK.
- You must maintain the mapping relationship between the CMK ID and the encrypted objects.

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.



- 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.

Notice

- 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.

2.1.1.3.7. 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.

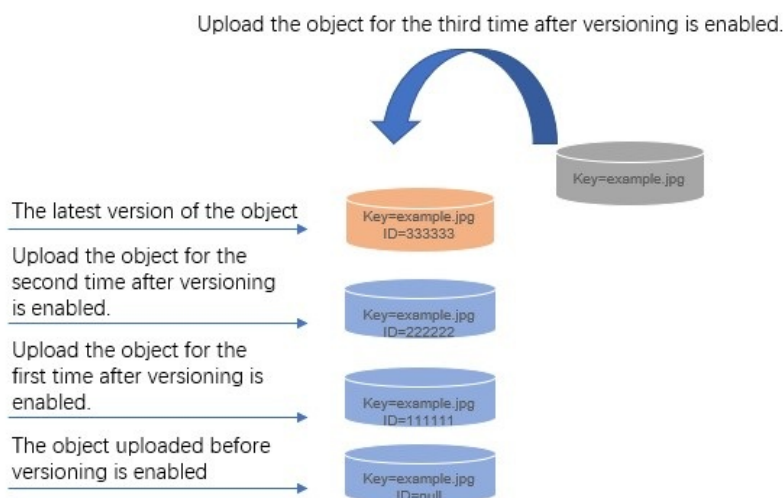
Versioning state	Object overwriting	Object deletion
Disabled	The existing object is overwritten and cannot be recovered. Only the current object version is retained in the bucket.	The object is deleted and cannot be accessed.
Enabled	A new version with a unique ID is generated for the object. The existing 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.

Versioning state	Object overwriting	Object deletion
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 overwritten 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.

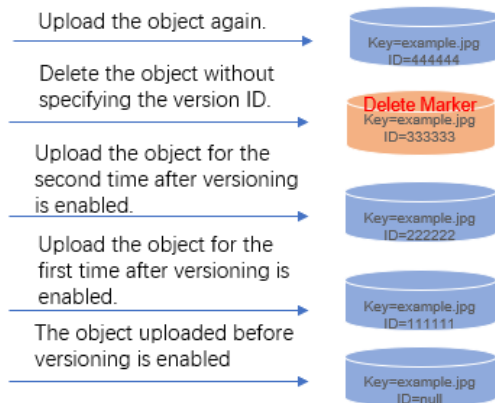
- Overwrite an object in a versioning-enabled bucket

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.



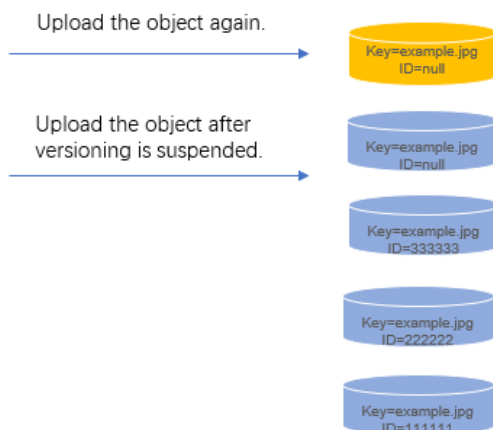
- Delete an object from a versioning-enabled bucket

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.



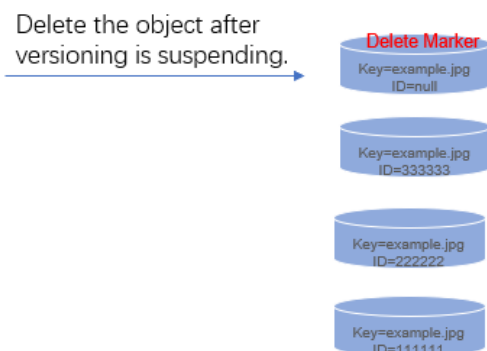
- Overwrite an object in a versioning-suspended bucket

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`

Parameter	Description	Valid value
<code>t</code>	Specifies the time when the image is to be captured.	[0, video duration] Unit: ms
<code>w</code>	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
<code>h</code>	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 <code>w</code> and <code>h</code> 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
<code>m</code>	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 <code>fast</code> , the most recent key frame before the specified time is captured.	<code>fast</code>
<code>f</code>	Specifies the format of the captured image.	<code>jpg</code> and <code>png</code>
<code>ar</code>	Specifies whether to automatically rotate the image based on the video information. If this parameter is set to <code>auto</code> , the system automatically rotates the image based on the video information.	<code>auto</code>

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

- Before you log on to the Apsara Uni-manager Management Console, the endpoint of the console is obtained from the deployment staff.
- We recommend that you use Google Chrome.

Procedure

- Enter the URL of the Apsara Uni-manager Management Console in the address bar and press the Enter key.
- 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** The first time that you log on to the Apsara Uni-manager Management Console, you must change the password of your account. For security purposes, your password must meet the minimum 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
- Special characters, including exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)

- Click **Log On**.
- If your account has multi-factor authentication (MFA) enabled, perform 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 **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 more information, see the *Bind a virtual MFA device to enable MFA* topic 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.

Procedure

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.  Note ◦ The region of a bucket cannot be changed after the bucket is created. ◦ If you want to access OSS from your ECS instance over the internal network, select the region where your ECS instance is deployed.
Cluster	Select a cluster for the bucket.
Bucket Name	Enter the name of the bucket.  Note ◦ 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.

Parameter	Description
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. Valid values: ◦ 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, see Modify 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. ◦ KMS: CMKs managed by KMS are used to encrypt objects in the bucket.
Encryption Algorithm	Specify an encryption algorithm. You can configure this parameter only when you set Encryption Method to OSS-Managed or KMS .
CMK	You can configure this parameter only when you set Encryption Method to KMS. OSS uses the specified CMK to encrypt objects in the bucket.  Note If no CMK is available, you must create one in the KSM console.

Parameter	Description
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.
KMS-based Encryption	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 that are encrypted by using CMKs managed by KMS to the destination bucket, you must enable KMS-based encryption, add a KMS-managed CMK for encryption and decryption, and attach a RAM role that has permissions to perform operations on objects and KMS-based encryption and decryption. You can configure this parameter only when Disable Zone-disaster recovery is set to Disable. To configure KMS-based encryption, perform the following steps: - Select the KMS-based Encryption check box. - Specify a CMK ID and a RAM role.
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.
RAM Role Name	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. ◦ 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.

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 a bucket](#).

Context

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.

Procedure

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.

Parameter	Description
Upload	Drag one or more objects to upload to this section, or click Upload to select one or more objects to upload.  Notice ◦ 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.

 **Notice** 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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket 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 a bucket](#).

Procedure

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. On the **Overview** tab, you can view the information about the bucket, which includes **Organization and Resource Set**, **Domain Names**, **Basic Settings**, and **VPC**.

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](#).



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

Procedure

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 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 a bucket](#).

Context

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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket 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 a bucket](#).

Procedure

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. 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 a bucket](#).

Context

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

After you configure a Referrer 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.

Procedure

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. 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 **Referrer 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 Referrer 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 Referrer 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 Referrer 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 Referrer 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 Referrer whitelist.
 - Select whether to turn on **Allow Empty Referrer** to allow requests in which the Referrer field is empty.

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

If you do not allow empty Referers fields, only HTTP or HTTPS requests which include an allowed Referrer 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 Referrer field and a request that does not contain the Referrer 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 Referrer whitelist but also allow empty Referrer fields. To preview a non-MP4 object by using the bucket domain name, you need only to allow empty Referrer 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 a bucket](#).

Context

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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket 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: <code>https://www.example.com:8080</code>. The following examples show how to configure domain names: ◦ To match a specified domain name, enter the full domain name. Example: <code>https://www.example.com</code>. ◦ Use an asterisk (*) as a wildcard in the domain name to match second-level domains. Example: <code>https://*.example.com</code>. ◦ 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.

Parameter	Required	Description
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 a bucket](#).

Context

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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket to which you want to upload objects.
3. Click the **Basic Settings** tab. Find the **Lifecycle** section. Click **Configure**.
4. 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 .  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 <code>img</code> , enter <code>img</code> .
	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 <code>img</code> , Key to <code>a</code> , and Value to <code>1</code> , the rule applies to all objects that have the <code>img</code> prefix in their names and have the tag <code>a=1</code> .
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.  Notice - 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 Date 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

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.  Notice 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 a bucket](#).

Context

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 quota 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. (Optional) 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.

 **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.

Context

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://`, `https://`, or `Aliyun`.
- 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 a bucket](#).

Context

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, see [Grants permissions to a role](#).

Procedure

1. Log on to the OSS console. For more information, see [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. 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.
RAM Role Name	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. ■ 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.

Context

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**.

 **Note** If you no longer need more object versions in a versioning-enabled bucket, you can select **Suspend**. In this case, OSS generates the 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**.

 Notice

- 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 [Manage 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.

Context

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.
4. 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**.

 **Notice** 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

Cross-region replication (CRR) allows you to perform automatic and asynchronous (near real-time) replication on objects across buckets that are located in different Object Storage Service (OSS) regions. If you enable CRR, operations such as the creation, overwriting, and deletion of objects can be synchronized from the source bucket to the destination bucket.

Prerequisites

The source bucket and destination bucket are created. For more information, see [Create a bucket](#).

Context

This feature meets the requirements of geo-disaster recovery or data replication. Objects in the destination bucket are extra replicas of objects in the source bucket. They have the same names, content, and metadata, such as the creation time, owner, user metadata, and object ACL.

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 CRR.
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	Specify the region in which the current bucket is located.
Source Bucket	Specify the name of the current bucket.
Destination Region	Specify the region in which the destination bucket is located. The source and destination buckets for CRR must be located in different regions. Data cannot be synchronized between buckets located within the same region.
Destination Bucket	Specify the destination bucket to which you want to synchronize data. The source bucket and destination bucket specified in a CRR rule are not allowed to synchronize data with other buckets. For example, if you configure a CRR rule to synchronize data from Bucket A to Bucket B, Bucket A and Bucket B are not allowed to synchronize data with other buckets.
Applied To	Specify the source data 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. For example, if you have a folder named <i>management/</i> in the root folder of a bucket and want to synchronize objects in a subfolder named <i>abc/</i> in <i>management/</i>, you can enter the prefix <i>management/abc/</i>. You can specify up to 10 prefixes.
Operations	Specify the synchronization policy. ◦ Add/Change: Only the added or updated 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.

Parameter	Description
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.  Notice When historical data is synchronized, objects in the destination bucket may be overwritten by historical data from the source bucket with the same name. Before you select this option, make sure that the data is consistent. ◦ No: Only objects that are uploaded or updated after CRR is enabled are synchronized to the destination bucket.

5. Click **Confirm**.

 Note

- It takes about 3 to 5 minutes to take effect after CRR is configured. Synchronization information is displayed after the source bucket is synchronized.
- In CRR, data is asynchronously replicated in near-real time. Depending on how much data needs to be replicated, it can take a few minutes to several hours to replicate the data to the destination bucket.

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**.
5. 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.

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**.
5. 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](#).
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 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.

Parameter	Description
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. Log management

2.4.16.1. 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 a bucket](#).

Procedure

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. 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**.

2.4.16.2. 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

Notice

- 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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket 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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket 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**.
4. 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.

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	Description
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.

Statement	Description
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

 **Notice** 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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket 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**.

2.4.17. Image processing

2.4.17.1. Configure image styles

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

Context

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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket 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,param_value1/action2,param_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%.

 **Note** 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.mytest-inc.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.mytest-inc.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**.

2.4.17.2. 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.

Context

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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket 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 <code>?x-oss-process=style/</code> 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 <code>http(s)://BucketName.Endpoint/ObjectName!StyleName</code> .

5. Click **Confirm**.

FAQ

- 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.

2.4.18. 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.
 - 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 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	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: <pre> { "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" }] }</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"
        ]
      }
    }
  ],
  "Version": "1"
}
```

- iii. Click **Save Changes**.

7. 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](#).

Context

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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket 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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket to which you want to upload objects.
3. Click the **Files** tab.
4. 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.
5. 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

Object Storage Service (OSS) uses object metadata to describe object attributes. Object metadata includes standard HTTP headers and user metadata. You can configure HTTP headers to customize HTTP request policies, such as cache policies and policies for forced object download. You can also configure user metadata to identify the purposes or attributes of objects.

Context

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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket 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 <i>image/gif</i> .
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: ◦ <i>identity</i>: OSS does not compress or encode the object. ◦ <i>gzip</i>: OSS uses the LZ77 compression algorithm created by Lempel and Ziv in 1977 and 32-bit cyclic redundancy check (CRC) to encode the object. ◦ <i>compress</i>: OSS uses the Lempel-Ziv-Welch (LZW) compression algorithm to encode the object. ◦ <i>deflate</i>: OSS uses the zlib library and the deflate algorithm to encode the object. ◦ <i>br</i>: OSS uses the Brotli algorithm to encode the object. For more information about Content-Encoding, see RFC 2616.  Notice 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 <code>Accept-Encoding: gzip</code> 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 <i>zh-CN</i> .

Parameter	Description
Content-Disposition	The method used to access the object. Valid values: ◦ <i>inline</i>: 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. ◦ <i>attachment</i>: The object is downloaded to the local computer. For example, if this header is set to <code>attachment; filename="example.jpg"</code>, the object is downloaded to the local computer. After the object is downloaded, the local file is named <code>example.jpg</code>. For more information about Content-Disposition, see RFC 2616.
Cache-Control	The cache configurations for the object. Valid values: ◦ <i>no-cache</i>: 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. ◦ <i>no-store</i>: All content of the object is not cached. ◦ <i>public</i>: All content of the object is cached. ◦ <i>private</i>: 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: <code>2022-10-12T00:00:00.000Z</code>. If <code>max-age=<seconds></code> is set for Cache-Control, <code>max-age=<seconds></code> takes precedence over Expires.
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 <code>x-oss-meta-</code> and assigned values. Example: <code>x-oss-meta-last-modified:20200909u</code>.

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 a bucket](#).

Context

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.

 **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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket to which you want to upload objects.
3. Click the **Files** tab. On the page that appears, click **Create Folder**.
4. 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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket 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.

Context

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.

 **Notice** Deleted objects cannot be recovered. Exercise caution when you delete objects.

Procedure

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. 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.

Context

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. Click **Buckets**. On the Buckets page that appears, click the name of the bucket 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.

Context

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.
5. 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.

Context

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

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 <code>abc/*</code>. ■ 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 <code>abc/myphoto.png</code>.
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: <code>arn:sts::{RoleOwnerUid}:assumed-role/{RoleName}/{RoleSessionName}</code>. For example, the role used to generate a temporary user is testrole, the UID of the Alibaba Cloud account that assumes the role is 12345, and the RoleSessionName that is specified when the temporary user is generated is testsession. In this case, enter <code>arn:sts::12345:assumed-role/testrole/testsession</code>. To grant all temporary users the permissions to access the specified resources, use asterisks (*) as wildcard characters. For example, enter <code>arn:sts::*:*/*/*</code>.  Notice 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.

Parameter	Description
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.  Notice If multiple bucket policies are configured for a user, the user has all the permissions defined in the policies. However, 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 the Read/Write permission to the same user, the permission of the user is Read/Write. If you configure a third policy to grant the None permission to the user, the permission of the user is None. - Advanced Settings If you select this option, you must 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.
Conditions	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. 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:

- Example 1: Allow anonymous users to list all objects in a bucket named examplebucket.

```
{
  "Statement": [
    {
      "Action": [
        "oss:ListObjects",
        "oss:ListObjectVersions"
      ],
      "Effect": "Allow",
      "Principal": [
        "*"
      ],
      "Resource": [
        "acs:oss:*:174649585760xxxx:examplebucket"
      ]
    }
  ],
  "Version": "1"
}
```

- Example 2: Forbid anonymous users whose IP addresses are not in the CIDR block 192.168.0.0/16 from managing a bucket named examplebucket.

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Deny",
      "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.

```

{
  "Statement": [
    {
      "Action": [
        "oss:GetObject",
        "oss:GetObjectAcl",
        "oss:GetObjectVersion",
        "oss:GetObjectVersionAcl"
      ],
      "Effect": "Allow",
      "Principal": [
        "20214760404935xxxx"
      ],
      "Resource": [
        "acs:oss:*:174649585760xxxx:examplebucket/hangzhou/2020/*",
        "acs:oss:*:174649585760xxxx:examplebucket/hangzhou/2015/*"
      ]
    },
    {
      "Action": [
        "oss:ListObjects",
        "oss:ListObjectVersions"
      ],
      "Condition": {
        "StringLike": {
          "oss:Prefix": [
            "hangzhou/2020/*",
            "hangzhou/2015/*"
          ]
        }
      },
      "Effect": "Allow",
      "Principal": [
        "20214760404935xxxx"
      ],
      "Resource": [
        "acs:oss:*:174649585760xxxx:examplebucket"
      ]
    }
  ],
  "Version": "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 a bucket](#).

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.

- **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 OSS and a virtual private cloud (VPC) to access OSS resources from the VPC.

Prerequisites

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

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.
Shared Scope	Yes	Select the scope of resources that you want to share. Valid values: Current Resource Set , Current Organization and Subordinate Organizations , and Current Organization . Default value: Current Resource Set.
Description	No	Enter the description of the single tunnel you want to create. The description cannot exceed 20 characters in length.

Parameter	Required	Description
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**.

After you create the single tunnel, you can click **Back to Console** in the pop-up dialog box to go back to the **Single Tunnel** page.

You can view the information about the single tunnel that you create on the **Single Tunnel** page.

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

2.8. CSG

2.8.1. What is CSG?

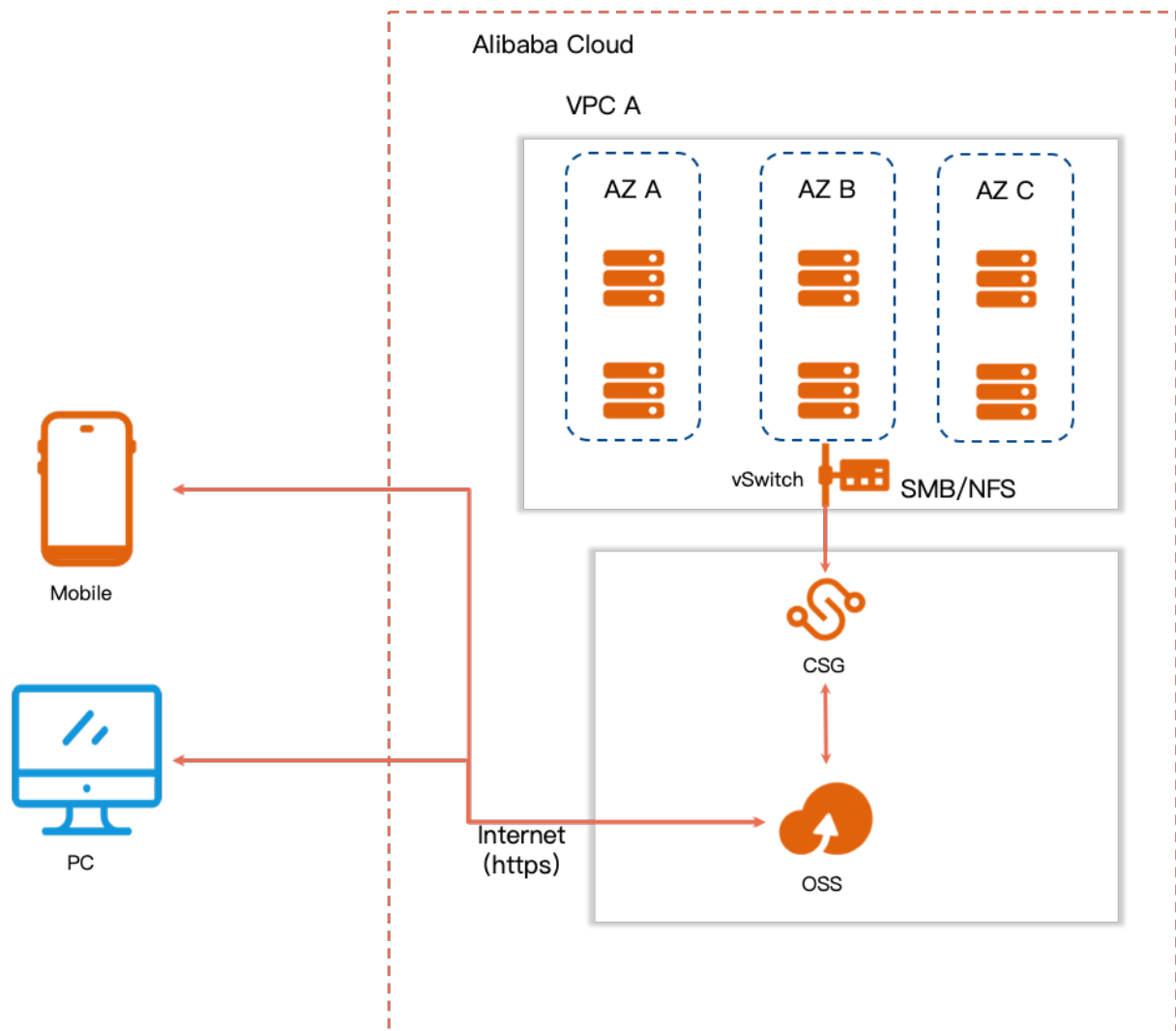
Cloud Storage Gateway (CSG) is a gateway service that can be deployed on Apsara Stack. CSG uses OSS buckets as backend storage devices. CSG provides on-premises and cloud applications with standard file services over the Network File System (NFS) and Server Message Block (SMB) protocols.

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.
- File 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, the file gateway retrieves the data of 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. 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*.

Context

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.  Notice You must select the VPC in which the ECS instance resides.
VSwitch	The vSwitch that is connected to the gateway.  Notice 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 OSS bucket.
Bucket AccessKey Secret	

 **Notice** 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 the Protocol 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 .

5. 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.

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](#).

2.8.4. File gateways

2.8.4.1. 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.  Notice You must select the VPC in which the ECS instance resides.

Parameter	Description
VSwitch	The vSwitch that is connected to the gateway.  Notice 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 user](#).

What's next

[Manage shares](#)

2.8.4.2. 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 OSS bucket.  Notice 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.
Bucket AccessKey Secret	

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 the Protocol 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 .

7. 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
User Mapping	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. ◦ 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.
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.

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 click **Settings** in the Actions column.
2. In the **SMB Share Settings** dialog box, set the parameters. The following table 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.

Parameter	Description
Read-only Users	The users who are allowed to only read data from the SMB share.  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 .
Delete a share	Find the share that you want to delete and click Delete in the Actions column.  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.

2.8.4.3. 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](#).

Context

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

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

2.8.4.4. 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.

2.8.4.5. Access shares

2.8.4.5.1. 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.

```
[root@centos7cb ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1       99G   1.6G   92G   2% /
devtmpfs        24G     0   24G   0% /dev
tmpfs           24G     0   24G   0% /dev/shm
tmpfs           24G  424K   24G   1% /run
tmpfs           24G     0   24G   0% /sys/fs/cgroup
tmpfs           4.8G     0   4.8G   0% /run/user/0
192.168.0.0:/nfs2 256T     0  256T   0% /mnt/nfs172cent7.4
[root@centos7cb ~]#
```

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.

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.

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:

- Run the `chkconfig netfs on` command to enable the netfs service to start at system startup.
- 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.d/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`: a local directory of the client. Specify a directory that supports read and write operations. You cannot specify a directory that does not exist.

- Run the `reboot` command to restart the ECS instance.

2.8.4.5.2. 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](#).

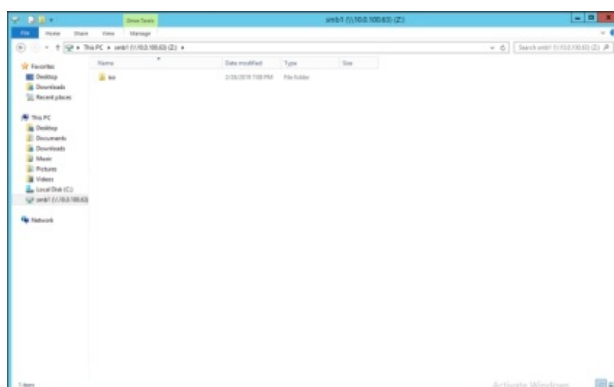
Note

- 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.



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](#).

 **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.

2.8.4.6. 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.

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

2.8.4.7. Appendix

2.8.4.7.1. 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:PutObject",  
    "oss:DeleteObject",  
    "oss:HeadObject",  
    "oss:CopyObject",  
    "oss:InitiateMultipartUpload",  
    "oss:UploadPart",  
    "oss:UploadPartCopy",  
    "oss:CompleteMultipartUpload",  
    "oss:AbortMultipartUpload",  
    "oss:ListMultipartUploads",  
    "oss:ListParts",  
    "oss:GetBucketStat",  
    "oss:GetBucketWebsite",  
    "oss:GetBucketInfo",  
    "oss:GetBucketEncryption",  
    "oss:GetBucketVersioning",  
    "oss:PutBucketEncryption",  
    "oss:DeleteBucketEncryption",  
    "oss:RestoreObject",  
    "oss:PutObjectTagging",  
    "oss:GetObjectTagging",  
    "oss:DeleteObjectTagging"  
]
```

3. NAS

3.1. What is NAS?

Apsara File Storage NAS is a cloud service that provides file storage solutions for compute nodes. The compute nodes include Elastic Compute Service (ECS) instances, instances, and Container Service for Kubernetes (ACK) clusters.

NAS provides standard file access protocols. You can use the distributed file storage solution that provides shared access, scalability, high reliability, and high performance without the need to modify the configurations of the applications. You can mount a NAS file system on multiple compute nodes at a time. This reduces a large number of costs in data transmission and synchronization.

You can perform the following operations:

- Create a NAS file system and mount target.
- Create a permission group for the file system and add rules to the permission group. This allows access from specific IP addresses or CIDR blocks to the file system. This also allows you to grant different levels of access permissions to the IP addresses or CIDR blocks.
- Mount the file system on compute nodes. The compute nodes include ECS instances and ACK clusters. NAS allows you to access the file system by using the Network File System (NFS) and Server Message Block (SMB) protocols. You can also call POSIX-based APIs to access the file system.
- Manage file systems, mount targets, and permission groups in the NAS console.
- Call NAS API operations to manage file systems.

3.2. Usage notes

Before you use NAS, take note of the following limits.

Limits on file systems

- Maximum number of files in each file system: 1 billion.
- Maximum length of a file system name: 255 bytes.
- Maximum size of each file: 32 TB.
- Maximum directory depth: 1,000 levels.
- Maximum capacity of a Performance NAS file system: 1 PB.
- Maximum number of compute nodes on which a file system can be mounted and from which the file system allows simultaneous access: 10,000.
- Maximum size of a protocol packet: 4 MB.
- Maximum number of Change Notify requests: 512.

Limits on NFS clients

The following list describes the limits on the usage of NFS clients:

- You can open up to 32,768 files at a time on an NFS client. Files in the list folder and its subfolders are not counted as part of the total number of open files.
- Each mount on an NFS client can obtain up to 8,192 locks across up to 256 files or processes. For example, a single process can obtain one or more locks on 256 separate files, or 8 processes can each obtain one or more locks on 32 files.
- We recommend that you do not use an NFS client in a Windows host to access an NFS file system.

Limits on SMB clients

Each file or folder can be opened up to 8,192 times in parallel across compute nodes that each have a file system mounted and users that share access to each of these file systems. This represents up to 8,192 active file handlers for each file system. A file system can have up to 65,536 active file handles.

Limits on the NFS protocol

- NAS supports the NFSv3 and NFSv4 protocols.
- NFSv4.0 does not support the following attributes: `FATTR4_MIMETYPE`, `FATTR4_QUOTA_AVAIL_HARD`, `FATTR4_QUOTA_AVAIL_SOFT`, `FATTR4_QUOTA_USED`, `FATTR4_TIME_BACKUP`, and `FATTR4_TIME_CREATE`. If one of the preceding attributes is applied to a file system, an `NFS4ERR_ATTRNOTSUPP` error appears on the client on which the file system is mounted.
- NFSv4.1 does not support the following attributes: `FATTR4_DIR_NOTIF_DELAY`, `FATTR4_DIR_NOTIF_DELAY`, `FATTR4_DACL`, `FATTR4_SACL`, `FATTR4_CHANGE_POLICY`, `FATTR4_FS_STATUS`, `FATTR4_LAYOUT_HINT`, `FATTR4_LAYOUT_TYPES`, `FATTR4_LAYOUT_ALIGNMENT`, `FATTR4_FS_LOCATIONS_INFO`, `FATTR4_MDSTHRESHOLD`, `FATTR4_RETENTION_GET`, `FATTR4_RETENTION_SET`, `FATTR4_RETENT_EVT_GET`, `FATTR4_RETENT_EVT_SET`, `FATTR4_RETENTION_HOLD`, `FATTR4_MODE_SET_MASKED`, and `FATTR4_FS_CHARSET_CAP`. If one of the preceding attributes is applied to a file system, an `NFS4ERR_ATTRNOTSUPP` error appears on the client on which the file system is mounted.
- NFSv4 does not support the following operations: `OP_DELEGPURGE`, `OP_DELEGRETURN`, and `NFS4_OP_OPENATTR`. If one of the preceding operations is applied to a file system, an `NFS4ERR_ATTRNOTSUPP` error appears on the client on which the file system is mounted.
- NFSv4 does not support delegations.
- UID and GID
 - On Linux, mappings between UIDs or GIDs and usernames or group names are defined in configuration files. For NFSv3 file systems, if the mapping between an ID and a name is defined in a configuration file, the name is displayed. If no mapping can be found for a UID or GID, the UID or GID is displayed.
 - For NFSv4 file systems, if the version of a Linux kernel is earlier than 3.0, the usernames and group names are displayed as `nobody` for all files. If the kernel version is later than 3.0, the rule used by NFSv3 file systems applies to display files.

 **Notice** If the Linux kernel version is earlier than 3.0, we recommend that you do not run the `chown` or `chgrp` command for files or directories in an NFSv4 file system. Otherwise, the UID and GID of the file or directory are changed to `nobody`.

Limits on the SMB protocol

- NAS supports the SMB 2.1 protocol and later versions. NAS also supports operating systems including Windows 7, Windows Server 2008 R2, and later versions. NAS does not support Windows Vista, Windows Server 2008, or earlier versions. Compared with SMB 2.1 and later versions, SMB 1.0 provides lower performance and fewer features. Windows products that support only SMB 1.0 have reached end of support.
- SMB file systems do not support extended file attributes and client-side caching based on leases.
- SMB file systems do not support I/O control (IOCTL) or file system control (FSCTL) operations. For example, you cannot create sparse files, compress files, check the status of network interface cards (NICs), or create reparse points.
- SMB file systems do not support alternate data streams.
- SMB file systems do not support identity authentication that is provided by Active Directory (AD) or Lightweight Directory Access Protocol (LDAP).
- SMB file systems do not support several features that are provided by SMB 3.0 or later, such as SMB Direct, SMB Multichannel, SMB Directory Leasing, and Persistent File Handle.
- SMB file systems do not support access control lists (ACLs) on files or directories.

Other limits

Before you delete a NAS file system, you must unmount the file system from the compute node on which you mounted the file system by using a mount target. Then, delete the mount target and the file system.

3.3. Quick start

3.3.1. Log on to the NAS console

This topic describes how to log on to the Apsara File Storage NAS console.

Prerequisites

- Before you log on to the Apsara Uni-manager Management Console, the endpoint of the console is obtained from the deployment staff.
- 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.
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** The first time that you log on to the Apsara Uni-manager Management Console, you must change the password of your account. For security purposes, your password must meet the minimum 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
- Special characters, including exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)

3. Click **Log On**.
4. If your account has multi-factor authentication (MFA) enabled, perform 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 **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 more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Management Console User Guide*.

5. In the top navigation bar, click **Products**. In the **Storage** section, click **Apsara File Storage NAS**.

3.3.2. Create a file system

This topic describes how to create a file system in the Apsara File Storage NAS console.

Context

Before you create a file system, take note of the following items:

- You can use an Alibaba Cloud account to create up to 1,000 file systems.
- The maximum capacity of a Performance NAS file system is 1 PB.

If you want to increase the maximum storage capacity, we recommend that you contact Alibaba Cloud technical support.

Procedure

1. Log on to the [NAS console](#).
2. In the left-side navigation pane, choose **File System > File System List**. On the page that appears, click **Create File System**.
3. In the **Create File System** panel, configure the parameters.

The following table describes the parameters.

Parameter	Description
Region	Select the region in which you want to create the file system.
Organization	Select the organization to which the file system belongs.
Resource Set	Select the resource set to which the file system belongs.
Zone	Select the zone where the file system resides.
Cluster	Select the cluster where the file system resides.
File System Name	The name of the file system. The name of each file system must comply with the following naming conventions: ◦ The name must be 2 to 128 characters in length. ◦ The name must start with a letter but cannot start with <code>http://</code> or <code>https://</code>. ◦ The name can contain letters, digits, colons (:), underscores (_), and hyphens (-).
Storage Type	Select Performance . The maximum capacity of a Performance NAS file system is 1 PB.
Protocol Type	The protocol type. Select NFS or SMB based on your business requirements. We recommend that you mount Network File System (NFS) file systems on Linux clients and Server Message Block (SMB) file systems on Windows clients.
Capacity	The capacity of the file system. Valid values: 0.5 to 10240. Unit: TB.

4. After you configure the preceding parameters, click **Submit**.

3.3.3. Create a permission group and add rules

This topic describes how to create a permission group and add rules to the permission group in the Apsara File Storage NAS console.

Context

In Apsara File Storage NAS, each permission group represents a whitelist. You can add rules to a permission group to allow access to a file system from specific IP addresses or CIDR blocks. You can also grant different access permissions to different IP addresses or CIDR blocks.

 **Note** You can use an Alibaba Cloud account to create up to 100 permission groups. If you want to increase the limit, we recommend that you contact Alibaba Cloud technical support.

Create a permission group

1. Log on to the [NAS console](#).
2. Choose **File System > Permission Group** and click **Create Permission Group**.
3. In the **Create Permission Group** dialog box, configure the parameters.

The following table describes the parameters.

Parameter	Description
Organization	The organization to which the permission group belongs.
Resource Set	The resource set to which the permission group belongs.
Region	The region in which you want to create the permission group.
Name	The name of the permission group. The name of each permission group must comply with the following naming conventions: ◦ The name must be 3 to 64 characters in length. ◦ The name can contain letters, digits, and hyphens (-).
Network Type	The network type. Valid values: Classic Network and VPC . Select a network type based on your business requirements.

4. After you configure the preceding parameters, click **OK**.

Create a rule

1. Log on to the [NAS console](#).
2. On the **Permission Group** page, find the permission group to which you want to add a rule and click **Manage**.
3. Click **Add Rules**.
4. In the **Add Rules** dialog box, configure the parameters.

Add rules
✕

* Authorized address ?

* Read and write permissions

* User permissions ?

* Priority ?

The following table describes the parameters.

Parameter	Description
Authorized Address	Specifies the authorized object to which the rule applies. You can specify an IP address or CIDR block. Only IP addresses are available for permission groups of the classic network type.
Read/Write Permissions	Specifies whether to allow read-only or read/write access to the file system from the authorized object. Valid values: Read-only and Read/Write .
User Permissions	Specifies whether to limit access from Linux to the file system. ◦ Do not limit root users (no_squash): allows access to a file system from root users. ◦ Limit root users (root_squash): denies access to a file system from root users. All root users are considered nobody users. ◦ Limit all users (all_squash): denies access to a file system from all users, including root users. All users are considered nobody users. The nobody user has the least permissions in Linux and can access only the public content of the file system. This ensures the security of the file system.
Priority	Specifies the priority of the rule. If multiple rules are applied to an authorized object, the rule that has the highest priority takes effect. Valid values: 1 to 100. A value of 1 indicates the highest priority.

5. Click **OK** to create the rule.

3.3.4. Create a mount target

After an Apsara File Storage NAS file system is created, you must create a mount target for the file system. Then, you can use the mount target to mount the file system on compute nodes. The compute nodes include Elastic Compute Service (ECS) instances and Container Service for Kubernetes (ACK) clusters. This topic describes how to create a mount target.

Prerequisites

The following operations are completed:

- **Create a file system** of application management in Kubernetes clusters
- **Create a permission group and add rules** of application management in ECS clusters

Context

A mount target is an endpoint that resides in a virtual private cloud (VPC) or the classic network. Each mount target corresponds to a file system. NAS file systems support mount targets of the VPC and classic network types.

Note You can use a mount target to mount a file system on multiple compute nodes for shared access. The compute nodes include ECS instances and ACK nodes.

Procedure

1. Log on to the **NAS console**.
2. Choose **NAS > File System List**.
3. Find the file system and click **Management** in the Actions column.
4. In the left-side navigation page, select **Mount Targets**.
5. On the **Mount Target** tab, click **Add Mount Target**.
6. In the **Add Mount Target** dialog box, configure the following parameters.

Mount Target Type: The type of the mount target. Valid values: **VPC** and **Classic Network**.

Note Mount targets of the classic network type allow access only from ECS instances that belong to the same Alibaba Cloud account as the mount targets.

- If you want to create a mount target in a VPC, configure the parameters that are described in the following table.

Parameter	Description

Parameter	Description
VPC Network	Select the VPC in which you want to create the mount target.  Note The VPC that you specified must be the same as the VPC in which the compute nodes reside. The compute nodes include ECS instances and ACK nodes.
Switch	Select a vSwitch that resides in the VPC.
Permission Group	Select an existing permission group.

- If you want to create a mount target in the classic network, configure the parameters that are described in the following table.

Parameter	Description
Permission Group	Select an existing permission group.

7. Click **OK** to create the mount target.

3.3.5. Mount an NFS file system

Before you mount a file system, you must create the file system and a mount target for the file system. Then, you can use the mount target to mount the file system on compute nodes. The compute nodes include Elastic Compute Service (ECS) instances and Container Service for Kubernetes (ACK) clusters. This topic describes how to mount a Network File System (NFS) file system.

Prerequisites

- A file system is created. For more information, see [Create a file system](#).
- A permission group is created and one or more rules are added to the permission group. For more information, see [Create a permission group and add rules](#).
- A mount target is created. In this example, a mount target of the VPC type is used. For more information, see [Add a mount target](#).
 - If the mount target of a file system resides in a virtual private cloud (VPC), you can mount the file system only on an ECS instance in the VPC. The IP address of the ECS instance must be added as an authorized address to a rule in the permission group that is attached to the mount target.
 - If the mount target of a file system resides in the classic network, you can mount the file system only on an ECS instance that belongs to the same Alibaba Cloud account as the mount target. The private IP address of the ECS instance must be added as an authorized address to a rule in the permission group that is attached to the mount target.
- A compute node is created. In this example, a Linux ECS instance is used.

Step 1: Install an NFS client

Before you mount an NFS file system on a Linux ECS instance, you must install an NFS client. If an NFS client is installed, skip this step.

1. Log on to the Linux ECS instance. For more information, see the *Connect to an instance* section of the *Get started* topic in ECS User Guide.
2. Install an NFS client.
 - If CentOS, Red Hat Enterprise Linux (RHEL), or Alibaba Cloud Linux is running on the ECS instance, run the following command to install the NFS client:

```
sudo yum install nfs-utils
```

- If Ubuntu or Debian is running on the ECS instance, run the following commands to install the NFS client:

```
sudo apt-get update
```

```
sudo apt-get install nfs-common
```

Step 2: Mount an NFS file system

1. Log on to the Linux ECS instance. For more information, see the *Connect to an instance* section of the Get started topic in ECS User Guide.
2. Mount the NFS file system.

Run the following command to mount the NFS file system. In the command, replace `file-system-id.region.nas.aliyuncs.com:/mnt` with an actual value.

- Use the NFSv4 protocol to mount a file system:

```
sudo mount -t nfs -o vers=4.0,minorversion=0,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport file-system-id.region.nas.aliyuncs.com:/mnt
```

- Use the NFSv3 protocol to mount a file system:

```
sudo mount -t nfs -o vers=3,nolock,proto=tcp,rsz=1048576,wsz=1048576,hard,timeo=600,retrans=2,noresvport file-system-id.region.nas.aliyuncs.com:/mnt
```

Mount parameters

Parameter	Description
file-system-id.region.nas.aliyuncs.com:/mnt	<Endpoint of the mount target>:<Name of the shared directory> <Path of the mount directory>. Replace the domain name of the mount target, the name of the shared directory, and the path of the mount directory with actual values. ▪ Endpoint of the mount target: You can log on to the NAS console, find the file system that you want to manage, and then click Manage to go to the Details page. Then, you can obtain the endpoint of the mount target of the file system on the Mounting Use tab. ▪ Directory of the NAS file system: the root directory (/) or a subdirectory, for example, /sub1. If a subdirectory is specified, make sure that the subdirectory exists. ▪ Path of the mount directory: the root directory / of the Linux ECS instance or a subdirectory, for example, /mnt. If the mount directory is a subdirectory, make sure that the subdirectory exists.
vers	The protocol version of the file system. Only NFSv3 and NFSv4 are supported. ▪ vers=3: uses NFSv3 to mount the file system. ▪ vers=4.0: uses NFSv4 to mount the file system.

Parameter	Description
Mount options.	If you want to mount a file system, you can specify multiple mount options. Separate multiple mount options with commas (,). For example, you can specify the following mount options: ▪ <i>rsiz</i>e: specifies the size of each data block that a client can read from the file system. Recommended value: 1048576. Unit: bytes. ▪ <i>wsiz</i>e: specifies the size of each data block that a client can write to the file system. Recommended value: 1048576. Unit: bytes.  Note To prevent performance degradation, we recommend that you specify the maximum value (1048576) for both the <i>rsiz</i>e mount option and the <i>wsiz</i>e mount option. ▪ <i>hard</i>: specifies that applications stop accessing a file system when the file system is unavailable, and wait until the file system is available. We recommend that you enable the <i>hard</i> parameter. ▪ <i>timeo</i>: specifies the period of time for which the NFS client waits before the client retries to send a request. Unit: deciseconds (tenths of a second). Recommended value: 600 (60 seconds).  Note If you need to modify the <i>timeo</i> mount option, we recommend that you specify 150 or a greater value. The <i>timeo</i> mount option is measured in deciseconds (tenths of a second). For example, the value 150 indicates 15 seconds. ▪ <i>retrans</i>: specifies the number of retries after the NFS client fails to send a request. Recommended value: 2. ▪ <i>noresvport</i>: specifies that a new TCP port is used to ensure network connectivity between the file system and the ECS instance when the network recovers from failure. We recommend that you enable the <i>noresvport</i> parameter.  Note When you configure parameters, take note of the following items: ▪ To prevent data inconsistency, we recommend that you do not use the <i>soft</i> mount option. Use of the <i>soft</i> option is at your own risk. ▪ We recommend that you use the default values for other mount options. A decrease in performance may occur due to changes in specific mount options. These mount options include the size of the read or write buffer or the use of attribute caching.

3. Run the `mount -l` command to view the mount result.

If the output that is similar to the following information appears, the mount is successful.

```
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
squashfs on /dev/squashfs type squashfs (rw,relatime)
debugfs on /dev/debugfs type debugfs (rw,relatime)
source on /var/lib/rfcs/rpc_pipefs type rpc_pipefs (rw,relatime)
shm on /dev/shm type tmpfs (rw,nosuid,nodev,noexec,relatime)
nfs on /mnt type nfs (rw,relatime,vers=4.0,rsize=1048576,wsz=1048576,namlen=255,hard,noreport_proto,tcp,lmso=680,retans=2,sec=sys,clientaddr=172.17.0.1,lock=none,addr=172.17.0.1,netdev)
```

After the file system is mounted, you can run the `df -h` command to view the size of the file system.

4. After the SMB file system is mounted, you can read data from and write data to the NAS file systems on the ECS instance.

You can access the file system the same way you access a local directory. The following figure shows an example.

```

[root@i75efow1gw421dnp0416q2 ~]# mkdir /mnt/dir1
[root@i75efow1gw421dnp0416q2 ~]# mkdir /mnt/dir2
[root@i75efow1gw421dnp0416q2 ~]# touch /mnt/file1
[root@i75efow1gw421dnp0416q2 ~]# echo 'some file content' > /mnt/file2
[root@i75efow1gw421dnp0416q2 ~]# ls /mnt
dir1 dir2 file1 file2 tmp

```

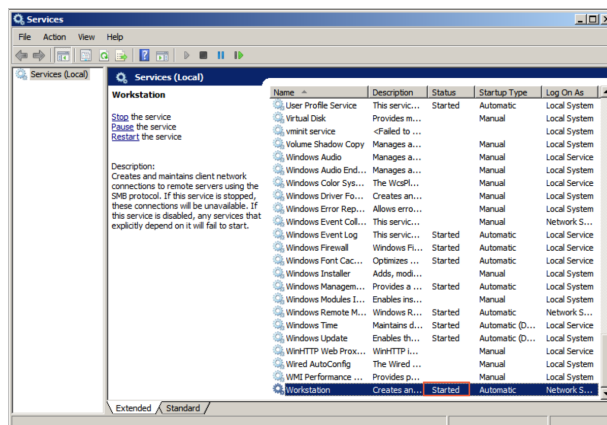
3.3.6. Mount an SMB file system

Before you mount a file system, you must create the file system and a mount target for the file system. Then, you can use the mount target to mount the file system on compute nodes such as Elastic Compute Service (ECS) instances. This topic describes how to mount a Server Message Block (SMB) file system.

Prerequisites

1. A file system is created. For more information, see [Create a file system](#).
2. A permission group is created and one or more rules are added to the permission group. For more information, see [Create a permission group and add rules](#).
3. A mount target is created. In this example, a mount target of the VPC type is used. For more information, see [Add a mount target](#).
 - If the mount target of a file system resides in a virtual private cloud (VPC), you can mount the file system only on an ECS instance in the VPC. The IP address of the ECS instance must be added as an authorized address to a rule in the permission group that is attached to the mount target.
 - If the mount target of a file system resides in the classic network, you can mount the file system only on an ECS instance that belongs to the same Alibaba Cloud account as the mount target. The private IP address of the ECS instance must be added as an authorized address to a rule in the permission group that is attached to the mount target.
4. An ECS instance is created. In this example, a Windows ECS instance is used.
5. The following services on the Windows host are started:
 - Workstation
 - a. Choose **All Programs > Accessories > Run**, or press **Win+R** and enter `services.msc` to go to the Services console.
 - b. Make sure that the Workstation service is in the **Started** state and the startup type is **Automatic**.

By default, the Workstation service is in the Started state.

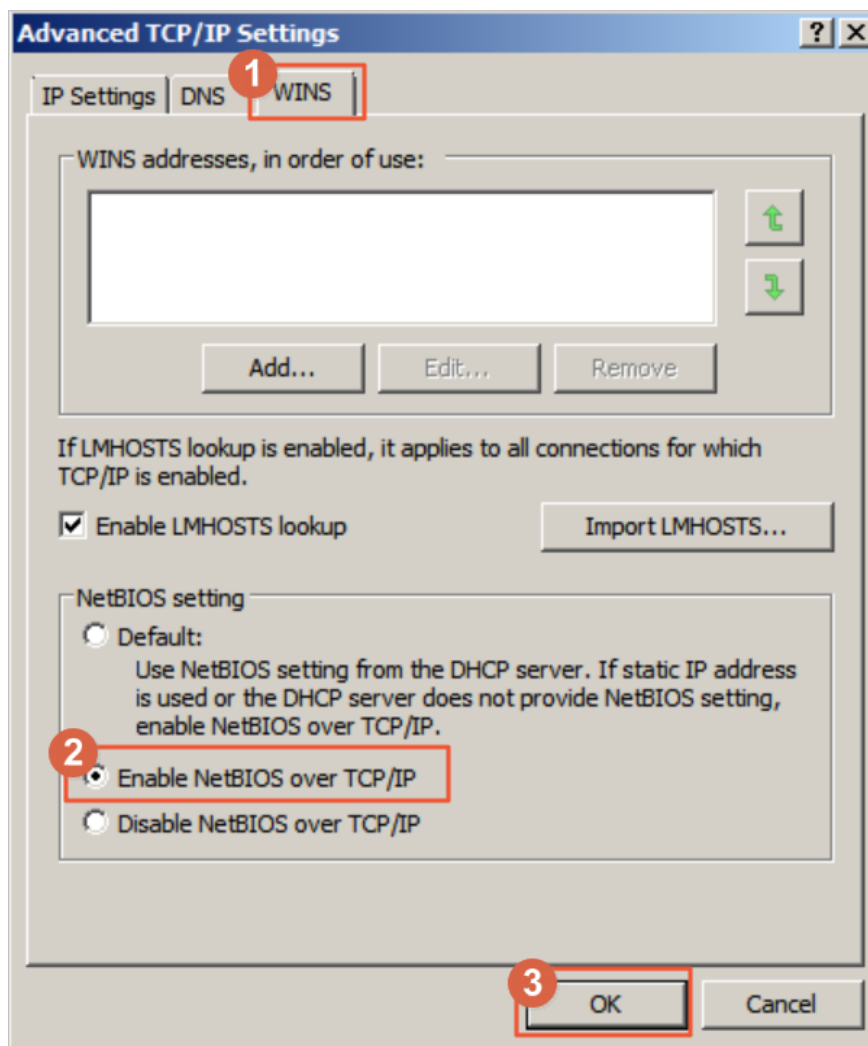


- TCP/IP NetBIOS Helper

To start the TCP/IP NetBIOS Helper service, perform the following steps:

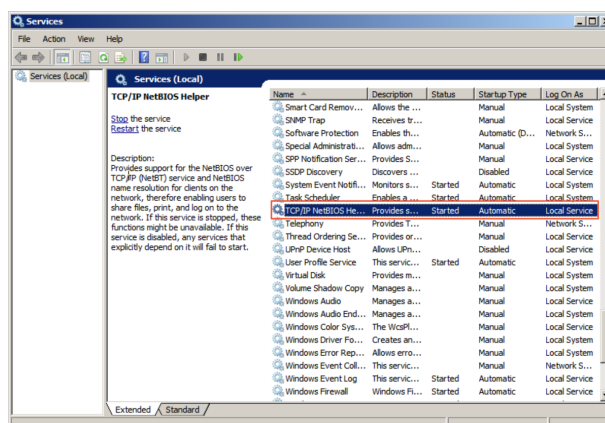
- a. On the **Network and Sharing Center** page, click the active network connection.
- b. Click **Properties** to open the Local Area Network Properties dialog box. Double-click **Internet Protocol Version 4 (TCP/IPv4)** to open the Internet Protocol Version 4 (TCP/IPv4) Properties dialog box, and then click **Advanced**.

- c. In the Advanced TCP/IP Settings dialog box, choose **WINS** > Enable NetBIOS over TCP/IP.



- d. Choose **All Programs > Accessories > Run**, or press **Win+R** and enter **services.msc** to go to the Services console.
- e. Make sure that the TCP/IP NetBIOS Helper service is in the **Started** state and the start up type is **Automatic**.

By default, the TCP/IP NetBIOS Helper service is in the Started state.



Procedure

1. Log on to the Windows ECS instance. For more information, see the *Connect to an instance* section of the *Get started* topic in ECS User Guide.
2. Open the Command Prompt and run the following command to mount an SMB file system:

```
net use D: \\file-system-id.region.nas.aliyuncs.com\myshare
```

The syntax of the command is `net use <the letter of a local drive> \\<the domain name of a mount target>\myshare`.

- Letter of the destination drive: the letter of the drive on which you want to mount an SMB file system. Replace the value of the parameter with the letter of an actual destination drive.

Note The letter of the destination drive must be different from the existing drive letters.

- The domain name of the mount target: When you create a mount target for a file system, the domain name of the mount target is automatically generated. You can replace the domain name based on your business requirements. To obtain the mount target of the file system, perform the following steps: Log on to the NAS console, find the file system that you want to manage, and then click **Manage** to go to the Details page.
 - myshare: the name of the shared SMB directory. You cannot change the name.
3. Run the `net use` command to check the mount result.

If the output that is similar to the following information appears, the mount is successful.



```
C:\Users\Administrator>net use
New connections will be remembered.

Status      Local        Remote                                           Network
-----
OK          D:           \\6778-2018-10-11-10-11-10.nas.aliyuncs.com\myshare  Microsoft Windows Network

The command completed successfully.
```

4. After the SMB file system is mounted, you can read data from and write data to the NAS file systems on the ECS instance.

3.4. File systems

3.4.1. View the details of a file system

This topic describes how to view the details of a file system. The details include the information about the file system and the attached mount targets. You can view the details of a file system in the Apsara File Storage NAS console.

Prerequisites

A file system is created. For information about how to detach a tag policy from an object, see [Create a file system](#).

Procedure

1. Log on to the [NAS console](#).
2. Choose **NAS > File System List**.
3. Find the file system and click **Management** to go to the details page of the file system.

The details page consists of the following sections:

- **Basic Information:** In this section, the information about the file system is displayed. The information includes the ID, region, protocol type, and storage specifications.
- **Mounting Targets:** In this section, the list of mount targets attached to the file system is displayed. You can manage the mount targets and view the clients on which the file system is mounted.

3.4.2. Delete a file system

This topic describes how to delete a file system in the Apsara File Storage NAS console.

Prerequisites

The number of mount targets is 0. For more information about how to view the number of mount targets, see [View the details of a file system](#).

Procedure

1. [Log on to the Apsara File Storage NAS console](#).
2. Choose **NAS > File System List**.
3. Find the file system that you want to manage, and click **Manage**.



Notice

- Before you can delete a file system, you must remove all mount targets from the file system.
- Proceed with caution when you delete a file system. After a file system is deleted, the data on the file system cannot be restored. Make sure that all data is backed up.

4. In the **Delete File System** dialog box, click **OK** to complete the deletion.

3.4.3. Scale up a file system

If the used space of an Apsara File Storage NAS file system reaches the specified capacity, data can no longer be written to the file system. To ensure the availability of the NAS service, we recommend that you scale up the file system before the used space reaches the specified capacity. This topic describes how to scale up a file system in the NAS console.

Prerequisites

A file system is created. For more information, see [Create a file system](#).



Notice A file system can be scaled up but cannot be scaled down. A Performance NAS file system can be scaled out to up to 1 PB. .

Procedure

1. Log on to the [NAS console](#).
2. In the left-side navigation pane, choose **File System > File System List**.
3. Find the file system and click **Upgrade** in the **Operations** column.
4. In the **Upgrade** dialog box, specify a desired capacity for the file system in the **New Total Capacity** field.
5. Click **OK**.

3.5. Mount targets

3.5.1. View mount targets

This topic describes how to view mount targets in the Apsara File Storage NAS console.

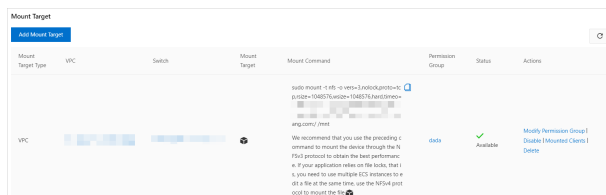
Context

The following operations are completed:

- **Create a file system** of application management in Kubernetes clusters
- **Create a permission group and add rules** of application management in ECS clusters
- **Optional. The update package is verified.**

Procedure

1. Log on to the **NAS console**.
2. Choose **NAS > File System List**.
3. Find the file system and click **Management** in the Actions column.
4. In the left-side navigation pane, click **Mounting Targets**. On the **Mounting Target** tab, view the mount targets of the file system.



3.5.2. Enable or disable a mount target

This topic describes how to enable or disable a mount target. You can control access to a mount target from clients by enabling or disabling the mount target.

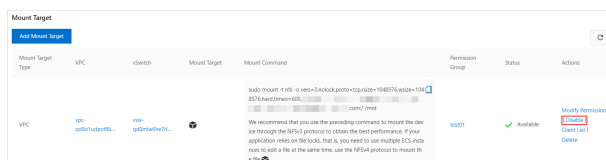
Prerequisites

The following operations are completed:

- **Create a file system** of application management in Kubernetes clusters
- **Create a permission group and add rules** of application management in ECS clusters
- **Optional. The update package is verified.**

Procedure

1. Log on to the **NAS console**.
2. Choose **NAS > File System List**.
3. Find the file system and click **Management** in the Actions column.
4. In the left-side navigation pane, click **Mounting Targets**. After you find the mount target that you want to disable or enable, you can perform the following operations:
 - To disable the mount target, click **Disable**. In the Disable Mount Target dialog box, click **OK** to deny access to the mount target from clients.
 - To enable the mount target, click **Enable**. In the Enable Mount Target dialog box, click **OK** to allow access to the mount target from clients.



3.5.3. Delete a mount target

This topic describes how to delete a mount target in the Apsara File Storage NAS console.

Prerequisites

A file system is unmounted from the compute node such as an Elastic Compute Service (ECS) instance and Container Service for Kubernetes (ACK) cluster on which you mounted the file system by using a mount target.

Procedure

1. Log on to the [NAS console](#).
2. Choose **NAS > File System List**.
3. Find the file system and click **Management** in the Actions column.
4. In the left-side navigation pane, click **Mounting Targets**.
5. Find the mount target that you want to delete and click **Delete**.

Notice Proceed with caution when you delete a mount target. After you delete a mount target, you cannot restore the mount target.

6. In the Delete Permission Group message, click **OK**.

3.5.4. Modify the permission group of a mount target

You must add a permission group to each mount target. You can modify the permission group of a mount target in the Apsara File Storage NAS console.

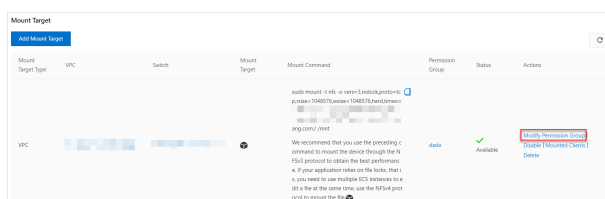
Prerequisites

The following operations are completed:

- [Create a file system](#) of application management in Kubernetes clusters
- [Create a permission group and add rules](#) of application management in ECS clusters
- [Optional. The update package is verified.](#)

Procedure

1. Log on to the [NAS console](#).
2. Choose **File System > File System List**.
3. Find the file system and click **Management** in the Actions column.
4. Click **Mounting Targets** to go to the **Mounting Target** tab.
5. Find the mount target that you want to modify and click **Modify Permission Group**.



6. In the **Modify Permission Group** dialog box, change the permission group and click **OK**.

3.6. Permission groups

3.6.1. View permission groups

This topic describes how to view permission groups in the Apsara File Storage console.

Prerequisites

A permission group is created. For information about how to detach a tag policy from an object, see [Create a permission group and add rules](#).

Procedure

1. Log on to the [NAS console](#).
2. Choose **File System > Permission Group** and view the permission groups.

Permission Group

General Purpose NAS

Create Permission Group

Permission Group Name / Description Enter

Permission Group Name	Organization	Resource Group	Type	Number of Rules	Number of Bound File Systems	Created At	Description	Actions
test01		ResourceSet(ap pstreaming)	VPC	0	1	Jul 8, 2022, 15:31:33	-	Manage Rules Edit Delete

Items per Page: 10 Total: 1 < Previous 1 Next >

3.6.2. Delete a permission group

This topic describes how to delete a permission group in the Apsara File Storage NAS console.

Prerequisites

A permission group is created. For more information, see [Create a permission group and add rules](#).

Operating system

1. [Log on to the Apsara File Storage NAS console](#).
2. Choose **NAS > Permission Group**.
3. Find the target permission group and click **Delete**.

Note Permission groups in use cannot be deleted. Before you can delete a permission group, you must remove the permission group from the linked mount target.

4. In the Delete Permission Group dialog box, click **OK**.

3.6.3. Manage permission group rules

This topic describes how to manage permission group rules in the Apsara File Storage NAS console. The management includes viewing the details of rules, modifying rules, and deleting rules.

Prerequisites

A permission group and a permission group rule are created. For more information, see [Create a permission group and add rules](#).

Procedure

1. [Log on to the Apsara File Storage NAS console](#).
2. Choose **NAS > Permission Group**.
3. Find the target permission group and click **Manage**.
4. On the Rules page, you can perform the following operations:
 - View all rules for the permission group.

Authorized address	Read and write permissions	User permissions	Priority	Operation
10.10.10.10	Read and write	All users are not anonymous (no, equal)	1	Edit Delete

Total 1 Previous Next 2

- Modify a rule. Find the target rule and click **Edit** to modify the details of the rule. The details include the **authorization address**, **read/write permissions**, **user permission**, and **priority**.
- Delete a rule. Find the target rule and click **Delete**. In the Delete Rule dialog box, click **OK**.

3.7. Directory-level ACLs that grant the read and write access

3.7.1. Overview

Apsara File Storage NAS supports NFSv4 access control lists (ACLs) and Portable Operating System Interface for UNIX (POSIX) ACLs. This topic introduces POSIX ACLs and NFSv4 ACLs and describes the usage notes for these ACLs.

Access control and user management are important for enterprise users who need to share files among different users and groups by using a shared file system. To control access to different files and directories, you can grant users and groups different types of access. NAS provides Network File System (NFS) ACLs to allow you to meet specific requirements. An ACL consists of one or more access control entries (ACEs) that each grant a user or group one or more permissions to access a file or directory.

The NFSv3 protocol implements access control by using the file mode creation mask (umask). The POSIX ACL feature is an extension to this permission model. You can grant permissions to specific users and groups in addition to the owner, group, and other classes. In a file system, a directory or file can inherit the permissions from its parent directory. For more information, see [acl\(5\) - Linux man page](#).

The NFSv4 ACL feature is an extension to the NFSv4 protocol. Compared with POSIX ACLs, NFSv4 ACLs provide finer-grained access control. For more information, see [nfs4_acl\(5\) - Linux man page](#).

You can use the NFSv3 protocol to mount a file system to which NFSv4 ACLs are applied. After the file system is mounted, the NFSv4 ACLs are converted into POSIX ACLs. You can also use the NFSv4 protocol to mount a file system to which POSIX ACLs are applied. After the file system is mounted, the POSIX ACLs are converted into NFSv4 ACLs. However, NFSv4 ACLs and POSIX ACLs are not fully compatible. The interaction between ACLs and the umask is not in an ideal state. NFSv3 file systems do not support locks. Therefore, if you use NFS ACLs, we recommend that you use the NFSv4 protocol to mount file systems and configure only NFS ACLs for the file systems. We also recommend that you do not use the umask or POSIX ACLs along with NFSv4 ACLs. For more information, see [Features](#).

Usage notes for POSIX ACLs

- We recommend that you use the default method that allows a subdirectory or file to inherit the ACL from the parent directory. Then, you do not need to configure another ACL when you create a file or subdirectory in the parent directory.
- A file system scans all ACEs each time it performs a permission check. To ensure optimal performance, we recommend that you retain a minimum number of ACEs. If redundant ACLs exist, the performance of the file system degrades.
- Proceed with caution when you configure ACLs by using the recursive method (`setfacl -R`). A large amount of metadata is generated when you apply an ACL recursively to a directory that contains a large number of files and subdirectories. This may negatively affect your business.
- Before you configure ACLs, we recommend that you manage groups and the corresponding permissions. For example, you can add a user to one or more groups. To add, remove, or change permissions for a user, you can move the user to a group that has the required permissions. You need to modify the ACL of the group only if the group structure changes. We recommend that you configure ACLs for groups rather than individual users. This is a simple and time-saving method to implement access control.

- You can apply a POSIX ACL to a user or group that accesses the file system from multiple clients. NAS identifies the user or group based on the user ID (UID) or group ID (GID). Therefore, you must set the same UID or GID for the user or group on the clients.
- All users have the permissions that are granted to the other class. Therefore, we recommend that you grant the least permissions to the other class. If the other class has more permissions than an ACE, a security vulnerability may exist.
- We recommended that you grant the least permissions to the other class. Before you create files or directories, run the `umask 777` command to configure the umask. This command sets the umask to 000 when a file or directory is being created. This ensures that the new file or directory has the least permissions. For more information about umask and the default mode, see [Why doesn't umask change execute permissions on files?](#)
- After you enable the POSIX ACL feature, the other class is moved to the everyone@ user category and the umask is also set. When NAS performs a permission check, NAS treats the other class as everyone.

Usage notes for NFSv4 ACLs

- We recommend that you use UIDs or GIDs such as UID 1001 to configure ACLs.
- We recommend that you do not configure the umask along with NFSv4 ACLs.
- The `nfs4_setfacl` command provides multiple options such as the -a, -x, and -m options to add, remove, and modify ACEs. We recommend that you run the `nfs4_setfacl -e <file>` command to edit an ACL in an editor.
- NAS NFSv4 ACLs support Allow ACEs rather than Deny ACEs. All users have the permissions that are granted to everyone. Therefore, we recommend that you grant the least permissions to everyone. If everyone has more permissions than an ACE, a security vulnerability may exist.
- NFSv4 ACLs support fine-grained permissions. In most cases, permission control based on fine-grained write permissions is unnecessary. For example, if a user has the write-data (w) access to a file but does not have the append-data (a) access, an error may occur when the user writes data to the file. The same issue occurs for a directory. To prevent unexpected permission errors, we recommend that you specify a capital W when you run the `nfs4_setfacl` command to configure an ACL. The `nfs4_setfacl` command specifies full write access. For a file, W is expanded to wadT. For a directory, W is expanded to wadTD.
- We recommend that you use the default method that allows a subdirectory or a file to inherit the ACL from the parent directory. Then, you do not need to configure another ACL when you create a file or subdirectory in the parent directory.
- A file system scans all ACEs each time it performs a permission check. To ensure optimal performance, we recommend that you retain a minimum number of ACEs. If redundant ACLs exist, the performance of the file system degrades.
- Proceed with caution when you configure ACLs by using the recursive method (`nfs4_setfacl -R`). A large amount of metadata is generated when you apply an ACL recursively to a directory that contains a large number of files and subdirectories. This may negatively affect your business.
- Before you configure ACLs, we recommend that you manage groups and the corresponding permissions. For example, you can add a user to one or more groups. To add, remove, or change permissions for a user, you can move the user to a group that has the required permissions. You need to modify the ACL of the group only if the group structure changes. We recommend that you configure ACLs for groups rather than individual users. This is a simple and time-saving method to implement access control.

3.7.2. Features

This topic describes the features of NFSv4 access control lists (ACLs) and POSIX ACLs.

Features of Apsara File Storage NAS NFSv4 ACLs

- Only access control entries (ACEs) of the allow type are supported. ACEs of the following types are not supported: deny, audit, and alarm.

Deny ACEs increase the complexity of access control. In most cases, complexity leads to confusion and increases potential security risks. As agreed by the industry, we recommend that you do not use deny ACEs. For more information about why deny ACEs are not recommended, see [FAQ](#).

Audit and alarm ACEs are not available for NFS file systems. However, you can audit file systems and configure alerts based on auditing results in the NAS console.

- If no ACL is set for a file or a directory, the default ACL that corresponds to the predefined umask is applied.

Example:

- i. Run the `touch file` command to enter the file.
- ii. Run the `ls -l file` command to view the permissions on the file.

```
-rw-r--r--. 1 root root 0 May  6 14:27 file
```

- iii. Run the `nfs4_getfacl file` command to view the current ACL permissions on the file.

```
# file: file
A::OWNER@:rwatTnNcCy
A::GROUP@:rtncy
A::EVERYONE@:rtncy
```

- An ACL is an ordered list that contains and deduplicates ACEs. This scheme ensures that the permissions defined in an ACL are clear and informative.

If you apply both a new ACE and an existing ACE to the same object and the existing ACE is inherited from the parent object, the permissions of the new ACE override the permissions of the existing ACE. Example:

- Run the `nfs4_getfacl file` command to view the ACL permissions on the file.

ACEs that include the principals OWNER, GROUP, and EVERYONE are queued in this sequence at the beginning of an ACL. These ACEs take precedence over other ACEs.

```
# file: file
A::OWNER@:rwaxTnNcCy
A::GROUP@:rtncy
A::EVERYONE@:rtncy
A::1001:rwaxTnNcCy
```

- Add an ACE of the read and write permissions to the following ACL for a user principal named 1009. The ACE is placed after the ACE that is defined for a user principal named 1001 based on the predefined order.

- Run the following commands:

```
nfs4_setfacl -a A::1009:X file
nfs4_getfacl file
```

- Sample command output:

```
# file: file
A::OWNER@:rwaxTnNcCy
A::GROUP@:rtncy
A::EVERYONE@:rtncy
A::1001:rwaxTnNcCy
A::1009:rwaxTnNcCy
```


- Add a new ACE that includes the execute permissions to the ACL for the user principal named 1009. The system automatically merges the execute permissions to the existing ACE for the 1009 user principal.

- Run the following commands:

```
nfs4_setfacl -a A::1009:W file
nfs4_getfacl file
```

- Sample command output:

```
# file: file
A::OWNER@:rwaxTcCy
A::GROUP@:rwatcy
A::EVERYONE@:tcy
A::1001:rwaxTTNcCy
A::1009:waxTncCy
```

- Add the f and d inheritance flags to an ACE that includes a user principal named 1009. Then, the system splits the ACE into two ACEs. One ACE has an extra inheritance flag named i, which indicates an inherit-only ACE. The other ACE applies to only the file object that does not have inheritance flags. If the inheritance type of an existing ACE matches the type of one of the two ACEs, the system combines the existing ACE with this ACE into one ACE.

- Run the following commands:

```
nfs4_setfacl -a A:fd:1009:R file
nfs4_getfacl file
```

- Sample command output:

```
# file: file
A::OWNER@:rwaxTcCy
A::GROUP@:rwatcy
A::EVERYONE@:tcy
A::1001:rwaxTTNcCy
A::1009:rwaxTTNcCy
A:fdi:1009:r
```

- All ACEs can be inherited.

- i. For example, the OWNER principal has the write permissions, the GROUP principal has the read permissions, and the EVERYONE has no permissions on the dir directory.

- Run the following commands:

```
nfs4_getfacl dir
```

- Sample command output:

```
# file: dir
A::OWNER@:rwaDxtTnNcCy
A::GROUP@:rxTcy
A::EVERYONE@:tnCy
```

- ii. Add an ACE that grants a user principal named 1000 the read, write, and execute permissions on the dir directory. The f and d inheritance flags are also specified for the ACE.

- Run the following commands:

```
nfs4_setfacl -a A:fd:1000:rwX dir
nfs4_getfacl dir
```

- Sample command output:

```
# file: dir
A::OWNER@:rwaDxtTcCy
A::GROUP@:rxtcy
A::EVERYONE@:tcy
A::1000:rwX
A:fdi:1000:rwX
```

- iii. When you create a file or subdirectory in the dir directory, the file or the subdirectory automatically inherits all ACEs from the dir directory.

- Create a file in the dir directory

- Run the following commands:

```
touch dir/file
nfs4_getfacl dir/file
```

- Sample command output:

```
# file: dir/file
A::OWNER@:rwaTcCy
A::GROUP@:rwaTcy
A::EVERYONE@:rwaTcy
A::1000:rwX
```

- Create a subdirectory in the dir directory

- Run the following commands:

```
mkdir dir/subdir
nfs4_getfacl dir/subdir
```

- Sample command output:

```
# file: dir/subdir
A::OWNER@:rwaDxtTcCy
A::GROUP@:rwaDxtcy
A::EVERYONE@:rwaDxtcy
A:fdi:1000:rwX
```

 Note

- We recommend that you grant the least permissions to the EVERYONE principal. Before you run one of the preceding commands, we recommend that you run the `umask 777` command. This command ensures that no access to a file or directory is granted when the file or directory is created. For more information about umask and the default mode, see [Why doesn't umask change execute permissions on files?](#)
- When Linux calls functions to create files or directories, the predefined umask is used as a request parameter. You can obtain the final ACL for a child object from the overlap of the inherited ACL (parent to child) and the umask, as specified in the [RFC 7530](#) standard. When you modify the group bits of a umask based on the standard, the permissions included in an ACL for each group must be less than or equal to the permissions defined in the group bits. However, this results in an invalid inheritance for groups. For example, if you want the subfile to inherit Group A:RWX but the predefined umask GROUPS:R is used as a request parameter, the ACE included in the ACL of Group A is then changed to Group A: R. In actual business scenarios, we recommend that you modify umasks only for ACLs that include the following principals: OWNER, GROUP, and EVERYONE. This prevents potential issues and ensures that the semantics are clear. To remove permissions from a group, we recommend that you remove the ACE that is associated with the group.

- You must manage the mappings between usernames or group names and user IDs (UIDs) or group IDs (GIDs) across multiple independent instances.

Apsara File Storage NAS NFS uses IP security groups instead of usernames to authenticate users. When you configure NFSv4 ACLs, UIDs or GIDs that are included in ACEs are stored in Linux. When you print an ACL for an object in a shell, Linux automatically loads the `/etc/passwd` file and converts UIDs or GIDs into usernames or group names. You must manage the mappings between usernames or group names and UIDs or GIDs across multiple instances. You must make sure that each username is mapped to its corresponding UID, and each group name is mapped to its corresponding GID.

- You can print NFSv4 ACL by using extended attributes.
 - Run the following commands:

```
getfattr -n system.nfs4_acl file
```

- Sample command output :

```
# file: file
system.nfs4_acl=0sAAAABgAAAAAAAAAAAAABYBhwAAAAZPV05FUkAAAAAAAAAAAAABIAhwAAAAZHUK9VUEAAAAAAAAAAAA
AABIAhwAAAA1FVkvSWU90RUAAAAAAAAAAAAAAAEAAAEEMTAwMAAAAAAAAAALAAAAwAAAAQxMDAwAAAAAAAAEAAAFgGQ
AAAABTEwMDAxAAAA
```

- You can use tools such as `cp` to migrate NFSv4 ACLs.

NAS allows you to migrate NFSv4 ACLs by using the `cp`, `tar`, and `rsync` tools. For more information, see [How to preserve NFS v4 ACLs via extended attributes when copying file](#).

The following `cp --preserve=xattr file1 file2` command copies the ACL of file1 when it copies file1 to file2. The `cp -ar dir1 dir2` command copies the ACL of file1 when it copies file1 to file2.

 **Note** NFSv4 ACLs may fail to be migrated if the version of the `rsync` tool is earlier than 3.1.2.

- Example 1: Migrate the ACL of a file.

- Run the `nfs4_getfacl file1` command to view the ACL permissions of the file1 file.

```
# file: file1
A::OWNER@:rwatTcCy
A::GROUP@:rwatcy
A::EVERYONE@:rwatcy
A::1000:rtncy
```

- Run the `cp --preserve=xattr file1 file2` command to copy the ACL of the file1 file to the ACL of the file2 file.

- Example 2: Migrate the ACL of a directory.

- Run the `nfs4_getfacl file2` command to view the ACL permissions of the file2 file.
- Run the `cp -ar dir1 dir2` command to copy the ACL of the dir1 directory to the dir2 directory.

```
# file: file2
A::OWNER@:rwatTcCy
A::GROUP@:rwatcy
A::EVERYONE@:rwatcy
A::1000:rtncy
```

- NFSv4 ACLs can interact with umasks. If you modify the ACL of an object, the umask of the object is changed. If you modify the umask of an object, the ACL of the object is also changed.

For example, the umask of the file object is 0666. The following example shows the file permissions and ACL permissions:

- Run the `ls -l file` command to view the permissions on the file.

```
-rw-rw-rw-. 1 root root 0 May  3  2019 file
```

- Run the `nfs4_getfacl file` command to view the ACL permissions of the file.

```
# file: file
A::OWNER@:rwatTcCy
A::GROUP@:rwatcy
A::EVERYONE@:rwatcy
```

- If you add the execute permissions to the umask by modifying the owner bits, the execute permissions are also added to the ACE that includes the OWNER principal. Example:

- Run the `chmod u+x file` command to grant the execute permissions to the OWNER principal.
- Run the `ls -l file` command to view the permissions on the file.

```
-rwxrw-rw-. 1 root root 0 May  3  2019 file
```

- Run the `nfs4_getfacl file` command to verify that the execute permissions are granted to the OWNER principal.

```
# file: file
A::OWNER@:rwxatTcCy
A::GROUP@:rwatcy
A::EVERYONE@:rwatcy
```

- If you add the execute permissions to an ACE that includes the GROUP principal, the execute permissions are also added to the umask.
 - Run the `nfs4_setfacl -a A::GROUP@:x file` command to grant the execute permissions to the GROUP principal.
 - Run the `ls -l file` command to view the permissions on the file.

```
-rwxrwxrw-. 1 root root 0 May  3  2019 file
```

? Note

- In the interactions between ACLs and umasks, the EVERYONE principal is equivalent to the other class. When you modify the other class, the change also applies to the EVERYONE principal. This operation has a slight impact on the semantics of permissions. For example, the current umask is 177. After you run the `chmod o+r` command, all users that include the file owner and group members have the read permissions. This is because the read permissions are added to the ACE that includes the EVERYONE principal. If no change is applied to the default umask, the owner and group classes still do not have the read permissions after you run the `chmod o+r` command.
- If no change is applied to NFSv4 ACLs, the semantics of the other class of the file mode remain unchanged. If you modify an NFSv4 ACL, the semantics of the other class change to the semantics of the EVERYONE principal and the latest semantics remain unchanged. We recommend that you do not use umasks together with NFSv4 ACLs.

- Interactions between NFSv4 ACLs and POSIX ACLs are supported.

You can mount NFSv3 file systems that have NFSv4 ACLs. These NFSv4 ACLs are then converted to POSIX ACLs. You can also mount NFSv4 file systems that have POSIX ACLs. These POSIX ACLs are then converted to NFSv4 ACLs.

 **Note** The semantics of POSIX ACLs are different from the semantics of NFSv4 ACLs. For example, the inheritance rules that apply to POSIX ACLs do not differentiate files or directories. NFSv4 ACLs have more permissions than POSIX ACLs. POSIX ACLs have only read, write, and execute permissions. To prevent potential issues, we recommend that you use either NFSv4 ACLs or POSIX ACLs.

If you configure an NFSv4 ACL for the `dir0` directory, the following permissions are required.

- Run the following commands:

```
sudo nfs4_getfacl dir0
```

- Sample command output:

```
A::OWNER@:tTnNcCy
A::GROUP@:tncy
A::EVERYONE@:tncy
A:fdi:EVERYONE@:tncy
A:fdi:OWNER@:tTnNcCy
A:fdi:GROUP@:tncy
A:g:19064:rxtncy
A:g:19065:rwaDxtTnNcCy
A:fdig:19064:rxtncy
A:fdig:19065:rwaDxtTnNcCy
```

If you configure a POSIX ACL for the `dir0` directory, the following permissions are required.

- Run the following commands:

```
sudo getfacl dir0
```

- Sample command output:

```
user::---
group::---
group:players:r-x
group:adminis:rwX
mask::rwX
other::---
default:user::---
default:group::---
default:group:players:r-x
default:group:adminis:rwX
default:mask::rwX
default:other::---
```

If you configure an NFSv4 ACL for the `dir0/file` file, the following permissions are required.

- Run the following commands:

```
sudo nfs4_getfacl dir0/file
```

- Sample command output:

```
A::OWNER@:tTnNcCy
A::GROUP@:tncy
A::EVERYONE@:tncy
A:g:19064:rxtncy
A:g:19065:rwaxtTnNcCy
```

If you configure a POSIX ACL for the `dir0/file` file, the following permissions are required.

- Run the following commands:

```
sudo getfacl dir0/file
```

- Sample command output :

```
user::---
group::---
group:players:r-x
group:adminis:rwX
mask::rwX
other::---
```

- The number of NFSv4 ACLs is limited.

By default, NAS supports a maximum of 100,000 different ACLs in each file system. Each ACL can contain a maximum of 500 ACEs.

 **Note** We recommend that you do not exceed the limits of ACLs or ACEs. Otherwise, more time and resources are consumed to verify permissions.

Features of NAS POSIX ACLs

- Permissions that are specified for users of the other class apply to everyone.

Everyone includes the owner, groups, and users that are related to each ACE. The other class is equal to the EVERYONE principal of an NFSv4 ACL.

 **Note** We recommend that you grant the least permissions to users of the other class.

For example, the following ACL is configured for the *myfile* file. Although the ACE contains a user named alice who does not have the write permissions, the write permissions propagate to the ACE because the permissions are specified for users of the other class.

```
# file: myfile
# owner: root
# group: root
user::rw-
user:alice:r--
group::r--
mask::r--
other::rw-
```

- Permissions that are configured by ACLs are not changed after you run the `chmod` command.

 **Note** We recommend that you do not modify the umask of a file to which a POSIX ACL is applied. You can configure permissions for the file by modifying the POSIX ACL.

- i. For example, an ACE that grants the players group the read and write permissions on the *myfile* file.

```
# file: myfile
# owner: root
# group: root
user::rw-
user:player:rw-
group::rw-
group:players:rw-
mask::rw-
other::---
```

- ii. The `chmod g-w myfile` or `chmod u-w myfile` command does not change the permissions that are granted to the player user or the players group. This is different from the [POSIX ACL standard](#). However, this ensures that the permissions granted by POSIX ACLs to non-reserved users are the same after you modify permissions by using the umask. Non-reserved users include all users except for the users of the owner, group, and other classes.

```
# file: myfile
# owner: root
# group: root
user::r--
user:player:rw-
group::r--
group:players:rw-
mask::rw-
other:---
```

- If the execute permissions are not granted to the group or other class of an ACL, the ACL does not have the execute permissions.

The rule is predefined in Linux. The execute action is allowed by the backend of NAS. However, to ensure that the execute permissions in the ACL are effective, you must grant the execute permissions to the group or other class.

For example, if the group and other classes do not have the execute permissions on the *myfile* file, the player user cannot execute the file. Example:

```
# file: myfile
# owner: root
# group: root
user::rw-
user:player:r-x
group::r--
mask::r-x
other::r--
```

If you grant the execute permissions to the group class, the execute permissions also propagate to the player user. Example:

```
# file: myfile
# owner: root
# group: root
user::rw-
user:player:r-x
group::r-x
mask::r-x
other::r--
```

- If you configure inheritable NFSv4 ACLs for directories that reside in NFSv3 file systems, these settings may not conform to the POSIX ACL standard.

Inheritance rules that apply to files are different from inheritance rules that apply to directories in NFSv4 ACLs. The same inheritance rules apply to both files and directories in POSIX ACLs.

 **Note** To prevent potential issues, we recommend that you apply either NFS4 ACLs or POSIX ACLs to an NFS file system.

- The umask value cannot be modified.

The umask of a POSIX ACL is generated by the permissions or operations of all users and groups. The umask has no practical meaning and cannot be changed.

- You must manage the mappings between usernames or group names and UIDs or GIDs across multiple instances.

Apsara File Storage NAS NFS uses IP security groups instead of usernames to authenticate users. When you configure POSIX ACLs, UIDs or GIDs that are included in ACEs are stored in Linux. When you print an ACL for an object in a shell, Linux automatically loads the `/etc/passwd` file and converts UIDs or GIDs to actual usernames or group names. You must manage the mappings between usernames or group names and UIDs or GIDs across multiple instances. You must make sure that each username is mapped to its corresponding UID, and each group name is mapped to its corresponding GID.

- You can print POSIX ACL by using extended attributes.

- Run the following commands:

```
getfattr -n system.posix_acl_access file
```

- Sample command output:

```
# file: file
system.posix_acl_access=0sAgAAAAEAAAD/////AgAFACAEAAAEAAAA/////xAABQD/////TAABAP/////8=
```

- You can use tools such as `cp` to migrate POSIX ACLs.

NAS allows you to migrate POSIX ACLs by using the `cp`, `tar`, and `rsync` tools. For more information, see [How to preserve NFS v4 ACLs via extended attributes when copying file](#).

The following `cp --preserve=xattr file1 file2` command copies the ACL of file1 when it copies file1 to file2. The `cp -ar dir1 dir2` command copies the ACL of file1 when it copies file1 to file2.

 **Note** POSIX ACLs may fail to be migrated if the version of the `rsync` tool is earlier than 3.1.2.

- Example 1: Migrate the ACL permissions of the file.

- Run the `getfacl file1` command to view the ACL permissions of the file1 file.

```
user:---
user:player:r-x
group:---
mask::r-x
other:--x
```

- Run the `cp --preserve=xattr file1 file2` command to copy the ACL of the file1 file to the ACL of the file2 file.

- Example 2: Migrate the ACL of a directory.

- Run the `getfacl file2` command to view the ACL permissions of the file2 file.

```
# file: file2
user:---
user:player:r-x
group:---
mask::r-x
other:--x
```

- Run the `cp -ar dir1 dir2` command to copy the ACL of the dir1 directory to the dir2 directory.

- The number of POSIX ACLs is limited.

By default, NAS supports a maximum of 100,000 different ACLs in each file system. Each ACL can contain a maximum of 500 ACEs.

 **Note** We recommend that you do not exceed the limits of ACLs or ACEs. Otherwise, more time and resources are consumed to verify permissions.

FAQ

Why are deny ACEs not supported?

- The position of an ACE that resides in an ACL is important.

The sequence for ACEs that reside in an NFSv4 ACL is random. A deny ACE may be placed in any position of an NFSv4 ACL. For example, an ACL contains two ACEs: A::Alice:r and D::Alice:r. The position of the ACEs determines whether the user named Alice has the write permissions.

 **Note** When you configure an ACL, you must consider the position of each ACE.

- The number of ACEs in an ACL has a sharp increase.

The ACEs in an ACL may be difficult to combine and deduplicate because it is not mandatory to sequence ACEs. The number of ACEs may increase up to tens or hundreds over a long period of time. To manage the final permissions that are produced by these ACEs, you need to check each ACE. The check process is strenuous and time-consuming.

- The interactions between the umask and ACLs become more complex after deny ACEs are applied. This is because deny features do not exist in the umask.
 - If deny ACEs are available, you may need to add several ACEs to an ACL when the umask is changed. For example, if you change the umask to -rw-rw-rw, you need to add the following ACEs to the ACL. You must add the ACEs in sequence at the beginning of the ACL.

```
A::OWNER@:rw
D::OWNER@:x
A::GROUP@:rw
D::GROUP@:x
A::EVERYONE@:rw
D::EVERYONE@:x
```

- If deny ACEs are unavailable, you can sort and deduplicate ACEs while the ACEs for the everyone class and the other class are not differentiated. If you change the umask, you can modify ACLs by changing ACEs for the owner, group, and everyone classes into the following content:

```
A::OWNER@:rw
A::GROUP@:rw
A::EVERYONE@:rw
```

- Conversions between NFSv4 ACLs and POSIX ACLs are not supported.

POSIX ACLs do not support deny ACEs. If deny ACEs are included in an NFSv4 ACL, you cannot convert the ACL to a POSIX ACL.

3.7.3. Use POSIX ACLs to control access

This topic describes how to configure Portable Operating System Interface (POSIX) access control lists (ACLs) to control access to the files and directories in an NFSv3 file system.

Prerequisites

An NFSv3 file system is mounted. For more information, see [Mount an NFS file system](#).

Description

Before you configure POSIX ACLs, we recommend that you get familiar with specific commands. The following table describes the commands.

Command	Description
<code>getfacl <filename></code>	Queries the ACL that applies to the specified file.
<code>setfacl -m g::w <filename></code>	Adds an access control entry (ACE) that grants the write permissions on a specified file to the GROUP@ principal.
<code>setfacl -m u:player:w <filename></code>	Grants the player user the write permissions.
<code>setfacl -m g:players:rwX <filename></code>	Grants the players group the read, write, and execute permissions.
<code>setfacl -x g:players <filename></code>	Revokes permissions from the players group
<code>getfacl file1 setfacl --set-file=- file2</code>	Copies the ACL for the <i>file1</i> file to the <i>file2</i> file.
<code>setfacl -b file1</code>	Removes all extended ACEs from the <i>file1</i> file. The base ACEs of the owner, group, and others are retained.
<code>setfacl -k file1</code>	Removes all default ACEs from the <i>file1</i> file.
<code>nfs4_setfacl -R -m g:players:rw dir</code>	Grants the players group the read and write permissions on the files and subdirectories in the <i>dir</i> directory.
<code>setfacl -d -m g:players:rw dir1</code>	Grants the players group the read and write permissions on the new files and subdirectories in the <i>dir1</i> directory.

Procedure

To control access to files and directories by configuring NFS ACLs, perform the following steps:

1. Create users and groups.

In this example, the following users are created: *player*, *admini*, and *anonym*. The following groups are created: *players* and *adminis*. The *player* user is added to the *players* group and the *admini* user is added to the *adminis* group.

```
sudo useradd player
sudo groupadd players
sudo usermod -g players player
sudo useradd admini
sudo groupadd adminis
sudo usermod -g adminis admini
sudo useradd anonym
```

2. Configure POSIX ACLs to control access to files and directories.

Run the following commands to complete the operations: Create a directory named *dir0* and grant the *players* group the read-only permissions, the *adminis* group the read, write, and execute permissions, and the *others* class no permissions on the files in the *dir0* directory.

```
sudo umask 777
sudo mkdir dir0
sudo setfacl -m g:players:r-x dir0
sudo setfacl -m g:adminis:rwX dir0
sudo setfacl -m u::--- dir0
sudo setfacl -m g::--x dir0
sudo setfacl -m o::--- dir0
sudo setfacl -d -m g:players:r-x dir0
sudo setfacl -d -m g:adminis:rwX dir0
sudo setfacl -d -m u::--- dir0
sudo setfacl -d -m g::--x dir0
sudo setfacl -d -m o::--- dir0
```

Run the `sudo getfacl dir0` command to verify the configuration.

```
# file: dir0
# owner: root
# group: root
user::---
group::--x
group:players:r-x
group:adminis:rwX
mask::rwX
other::---
default:user::---
default:group::--x
default:group:players:r-x
default:group:adminis:rwX
default:mask::rwX
default:other::---
```

3. Verify the configuration of the ACL.

- i. Run the following commands to verify that the admini user has the read and write permissions.

```
sudo su admini -c 'touch dir0/file'
sudo su admini -c 'echo 123 > dir0/file'
```

ii. Run the following command to verify that the player user has the read-only permissions.

- Run the `sudo su player -c 'touch dir0/file'` command to create a file named *file* in the *dir0* directory.

If the output is similar to the following information, the player user does not have the permissions to create the *file* file.

```
touch: cannot touch 'dir0/file': Permission denied
```

- Run the `sudo su player -c 'cat dir0/file'` command to view the content of the *dir0/file* file.

```
123
```

- Run the `sudo su player -c 'echo 456 >> dir0/file'` command to append data to the *dir0/file* file.

```
bash: dir0/file: Permission denied
```

- Run the `sudo su player -c 'getfacl dir0/file'` command to view the permissions that the player user has on the *dir0/file* file.

```
# file: dir0/file
# owner: admini
# group: adminis
user::---
group::---
group:players:r-x
group:adminis:rw-
mask::rw-
other::---
```

iii. Run the following commands to verify that the anonym user does not have permissions on specific resources:

- Run the `sudo su anonym -c 'ls dir0'` command to view the files in the *dir0* directory.

If the output is similar to the following information, the anonym user does not have permissions on the *dir0* directory.

```
ls: cannot open directory dir0: Permission denied
```

- Run the `sudo su anonym -c 'cat dir0/file'` command to view the content of the file.

If the output is similar to the following information, the anonym user does not have the permissions to view the content of the file.

```
cat: dir0/file: Permission denied
```

- Run the `sudo su anonym -c 'getfacl dir0/file'` command to query the permissions that the anonym user has on the file.

If the output is similar to the following information, the anonym user does not have permissions on the file.

```
getfacl: dir0/file: Permission denied
```

What to do next

You can use the following method to revoke permissions from a user.

If you want to use NFSv4 ACLs, we recommend that you add each user to groups. This allows you to configure permissions for a group rather than a separate user. To revoke permissions on an object from a user, you can remove the user from the group that is granted permissions on the object. For example, run the following commands to remove the admini user from the adminis group and add the user to the adminis2 group:

1. Run the `sudo groupadd adminis2` command to create a group named adminis2.
2. Run the `sudo usermod -g adminis2 admini` command to remove the admini user from the adminis group and add the user to the adminis2 group.
3. Run the `id admini` command to query the ID information about the admini user.

```
uid=1057(admini) gid=1057(admini) groups=1061(adminis2)
```

4. Run the following commands to check the permissions that the admini user has:
 - Run the `sudo su admini -c 'ls dir0'` command. If the output is similar to the following information, the admini user does not have permissions on the `dir0` directory.

```
ls: cannot open directory dir0: Permission denied
```

- Run the `sudo su admini -c 'cat dir0/file'` command. If the output is similar to the following information, the admini user does not have the permissions to view the content of the `dir0/file` file.

```
cat: dir0/file: Permission denied
```

- Run the `sudo su admini -c 'getfacl dir0/file'` command. If the output is similar to the following information, the admini user does not have permissions on the `dir0/file` file.

```
getfacl: dir0/file: Permission denied
```

3.7.4. Use NFSv4 ACLs to control access

This topic describes how to configure NFSv4 access control lists (ACLs) and use the ACLs to control access to the files or directories in an NFSv4 file system.

Prerequisites

An NFSv4 file system is mounted. For more information, see [Mount an NFS file system](#).

Context

You can mount an NFSv4 file system on a Linux Elastic Compute Service (ECS) instance and install the Linux-specific `nfs4-acl-tools` tool on the instance. After the tool is installed, you can use the standard `nfs4_getfacl` and `nfs4_setfacl` tools to configure NFSv4 ACLs.

Description

Before you configure NFSv4 ACLs, we recommend that you get familiar with specific commands. The following table describes the commands.

Command	Description
<code>nfs4_getfacl <filename></code>	Views the ACL that applies to a specified file
<code>nfs4_setfacl -a A::GROUP@:W <filename></code>	Adds an access control entry (ACE) that grants the write permissions on a specified file to the GROUP@ principal.
<code>nfs4_setfacl -a A::1000:W <filename></code>	Adds an ACE that grants the write permissions on a specified file to a user principal named 1000.
<code>nfs4_setfacl -a A:g:10001:W <filename></code>	Adds an ACE that grants the write permissions on a specified file to a user principal named 10001.
<code>nfs4_setfacl -e <filename></code>	Modifies an ACL in interactive mode.
<code>nfs4_getfacl <filename> > saved_acl.txt</code>	Saves a list of permissions on a specified file as a TXT file.

Command	Description
<code>nfs4_setfacl -S saved_acl.txt <filename></code>	Configures permissions on a specified file by using a TXT file that includes a list of permissions.
<code>nfs4_setfacl -m A::1001:rwaxTNCy A::1001:rxtcy file1</code>	Modifies an ACE that applies to the <i>file1</i> file.
<code>nfs4_getfacl file1 nfs4_setfacl -S - file2</code>	Copies the ACL for the <i>file1</i> file to the <i>file2</i> file.
<code>nfs4_getfacl file1 grep @ nfs4_setfacl -S - file1</code>	Removes all ACEs that apply to the <i>file1</i> file except for ACEs that include the following principals: OWNER@, GROUP@, and EVERYONE@.
<code>nfs4_setfacl -R -a A:g:10001:rW dir</code>	Adds an ACE that grants the read and write permissions on the files and subdirectories in the <i>dir</i> directory to a group principal named 10001.
<code>find dir -type f -exec sh -c 'for ace in \$(nfs4_getfacl \{} grep "^A.*\:1005\:"); do nfs4_setfacl -x \$ace \{}; done' \;</code>	Removes ACEs that grant permissions on the files in the <i>dir</i> directory to a user principal named 1005.
<code>nfs4_setfacl -a A:fdg:10001:rW dir1</code>	Adds an ACE that grants the read and write permissions on new files and subdirectories in the <i>dir1</i> directory to a group principal named 10001.
<code>nfs4_setfacl -a A:fg:10001:rx dir1</code>	Adds an ACE that grants the read, write, and execute permissions on all new files in the <i>dir1</i> directory to a group principal named 10001.

Procedure

To control access to files and directories by configuring NFSv4 ACLs, perform the following steps.

1. Create users and groups.

In this example, the following users are created: *player*, *admini*, and *anonym*. The following groups are created: *players* and *adminis*. The *player* user is added to the *players* group and the *admini* user is added to the *adminis* group.

```
sudo useradd player
sudo groupadd players
sudo usermod -g players player
sudo useradd admini
sudo groupadd adminis
sudo usermod -g adminis admini
sudo useradd anonym
```

2. Install the required tools to configure NFSv4 ACLs.

If you have installed these tools, skip this step.

```
sudo yum -y install nfs4-acl-tools
```

3. Obtain the IDs of the *players* and *adminis* groups.

Open the */etc/group* file. The IDs of the *players* and *adminis* groups are displayed.

```
players:x:19064:player
adminis:x:19065:admini
```

4. Configure NFSv4 ACLs to control access to files and directories.

Run the following commands to complete the operations: Create a directory named *dir0* and add ACEs that grant the read-only permissions on all files in the *dir0* directory to the players group, grant the read, write, and execute permissions to the adminis group, and do not grant permissions to other users.

```
sudo umask 777
sudo mkdir dir0
sudo nfs4_setfacl -a A:fdg:19064:RX dir0
sudo nfs4_setfacl -a A:fdg:19065:RWX dir0
sudo nfs4_setfacl -a A:fdg:OWNER@: dir0
sudo nfs4_setfacl -a A:fdg:GROUP@: dir0
sudo nfs4_setfacl -a A:fdg:EVERYONE@: dir0
```

Run the `sudo nfs4_getfacl dir0` command to verify the configuration.

```
A::OWNER@:tTnNcCy
A::GROUP@:tnCy
A::EVERYONE@:tnCy
A:fdi:EVERYONE@:tnCy
A:fdi:OWNER@:tTnNcCy
A:fdi:GROUP@:tnCy
A:g:19064:rxtncy
A:g:19065:rwaDxtTnNcCy
A:fdig:19064:rxtncy
A:fdig:19065:rwaDxtTnNcCy
```

5. Verify the configuration of the ACL.

- i. Run the following commands to verify that the admini user has the read and write permissions.

```
sudo su admini -c 'touch dir0/file'
sudo su admini -c 'echo 123 > dir0/file'
```

- ii. Run the following command to verify that the player user has the read-only permissions.

- Run the `sudo su player -c 'touch dir0/file'` command. If the output is similar to the following information, the player user does not have the permissions to create the *dir0/file* file.

```
touch: cannot touch 'dir0/file': Permission denied
```

- Run the `sudo su player -c 'echo 456 >> dir0/file'` command. If the output is similar to the following information, the player user does not have the write permissions on the *dir0/file* file.

```
bash: dir0/file: Permission denied
```

- Run the `sudo su player -c 'cat dir0/file'` command. If the output is similar to the following information, the player user does not have the permissions to view the content of the *dir0/file* file.

```
123
```

- Run the `sudo su player -c 'nfs4_getfacl dir0/file'` command to view the permissions that the player user has on the *dir0/file* file.

```
A::OWNER@:tTnNcCy
A::GROUP@:tnCy
A::EVERYONE@:tnCy
A:g:19064:rxtncy
A:g:19065:rwaxtTnNcCy
```

- iii. Run the following command to verify that the anonym user does not have permissions on specific resources:

- Run the `sudo su anonym -c 'ls dir0'` command. If the output is similar to the following information, the anonym user does not have permissions on the *dir0* directory.

```
ls: cannot open directory dir0: Permission denied
```

- Run the `sudo su anonym -c 'cat dir0/file'` command. If the output is similar to the following information, the anonym user does not have the permissions to view the content of the *dir0/file* file.

```
cat: dir0/file: Permission denied
```

- Run the `sudo su anonym -c 'nfs4_getfacl dir0/file'` command. If the output is similar to the following information, the anonym user does not have permissions on the *dir0/file* file.

```
Invalid filename: dir0/file
```

What to do next

You can use the following method to revoke permissions from a user.

If you want to use NFSv4 ACLs, we recommend that you add each user to groups. This allows you to configure permissions for a group rather than a separate user. To revoke permissions on an object from a user, you can remove the user from the group that is granted permissions on the object. For example, you can run the following commands to remove the admini user from the adminis group and add the user to the adminis2 group:

1. Run the `sudo groupadd adminis2` command to create a group named adminis2.
2. Run the `sudo usermod -g adminis2 admini` command to remove the admini user from the adminis group and add the user to the adminis2 group.
3. Run the `id admini` command to view the ID information about the admini user.

```
uid=1057(admini) gid=1057(admini) groups=1054(adminis2)
```

4. Run the following commands to check the permissions that the admini user has:

- Run the `sudo su admini -c 'ls dir0'` command. If the output that is similar to the following information appears, the admini user does not have permissions on the *dir0* directory.

```
ls: cannot open directory dir0: Permission denied
```

- Run the `sudo su admini -c 'cat dir0/file'` command. If the output that is similar to the following information appears, the admini user does not have the permissions to view the content of the *dir0/file* file.

```
cat: dir0/file: Permission denied
```

- Run the `sudo su admini -c 'nfs4_getfacl dir0/file'` command. If the output that is similar to the following information appears, the admini user does not have permissions on the *dir0/file* file.

```
Invalid filename: dir0/file
```


4. EBS

4.1. 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.

 Notice

- 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	$\min(1,800 + 30 \times \text{Capacity}, 25,000)$	$\min(1,800 + 8 \times \text{Capacity}, 5,000)$	$\min(1,800 + 30 \times \text{Capacity}, 25,000)$	$\min(1,800 + 8 \times \text{Capacity}, 5,000)$
Formula for calculating the throughput per disk (MB/s)	$\min(120 + 0.5 \times \text{Capacity}, 300)$	$\min(100 + 0.15 \times \text{Capacity}, 140)$	$\min(120 + 0.5 \times \text{Capacity}, 300)$	$\min(100 + 0.15 \times \text{Capacity}, 140)$
API parameter value	cloud_ppperf	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

 **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.

4.1.1. Features

4.1.1.1. 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 as a batch by using the group.

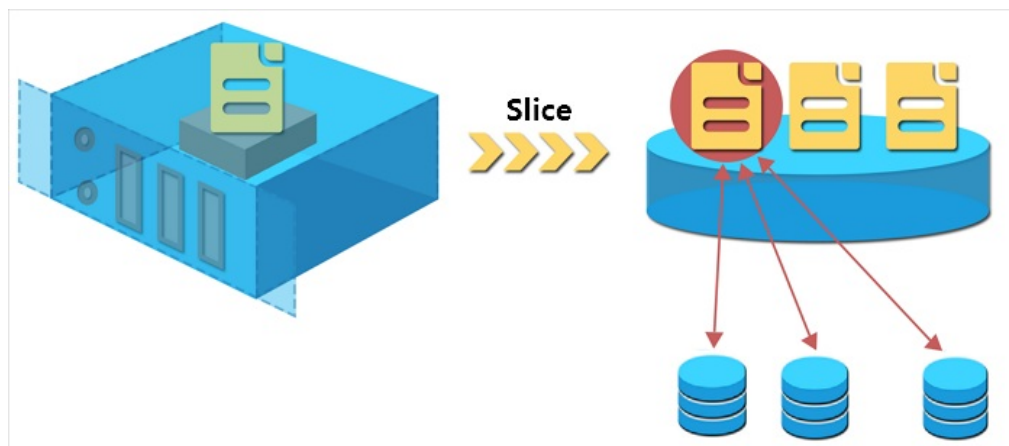
4.1.1.2. Triplicate storage

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.

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:

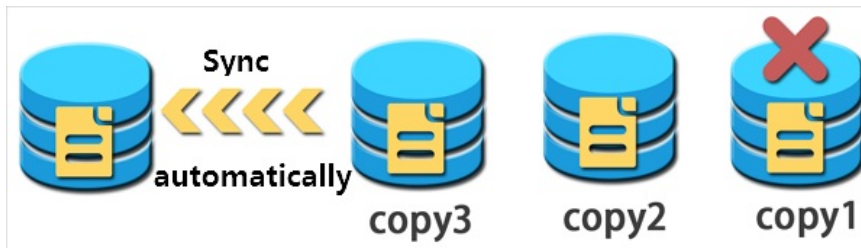
1. The client determines the location of a chunk corresponding to the write operation.
2. The client sends a request to the master to query the chunk servers where the three chunk replicas are each stored.
3. The client sends write requests to the chunk servers based on the results returned from the 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 ensures 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.

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.

4.1.1.3. 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	$\frac{m}{m+n}$ 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.

4.1.1.4. 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.

4.1.1.5. 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 <i>CDS User Guide</i> .
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 <code>RebootInstance</code> operation for the new size to take effect.	For more information, see the <i>Resize disks</i> topic under EBS of <i>CDS User Guide</i> .

- 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.

4.1.1.6. Snapshots

4.1.1.6.1. 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:

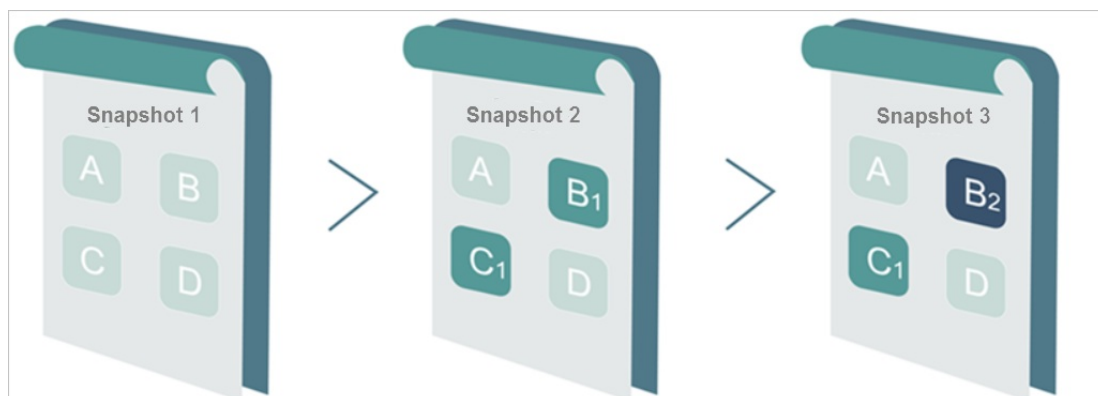
- You may want to use the data of one disk as the basis to write or store data to a different disk. To achieve this, you can create a snapshot for a cloud disk and then create another cloud disk from the snapshot. The new disk contains the basic data of the original disk.
- While cloud disks are a secure 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.

4.1.1.6.2. 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](#).

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.

 **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.

4.1.1.6.3. 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.

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.

4.1.1.7. 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
Async replication	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.
	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.
	Create a replication pair-consistent group	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. 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.

Feature	Operation	Description
Replication pair-consistent group	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.

Term	Description
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
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	✓	✓	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.

Item	Method	Support for primary disks	Support for secondary disks	Description
Use a system disk	Use the ECS console	✓	✓	- 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	✓	×	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	✓	✓	After a disk is deleted for a period of time, the state of the replication pair to which the disk belongs changes to Invalid .
Initialize a disk	ReinitDisk	✓	✓	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 console - ReinitDisk	✓	✓	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 console - ResetDisk	×	×	- 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 console - ReplaceSystemDisk	✓	✓	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	✓	✓	After you resize a disk online, the state of the replication pair to which the disk belongs changes to Invalid .

Item	Method	Support for primary disks	Support for secondary disks	Description
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.  Notice 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	×	×	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.

4.2. 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 use to log on to the Apsara Uni-manager Management Console from the operations administrator.

Note The first time that you log on to the Apsara Uni-manager Management Console, you must change the password of your account. For security purposes, your password must meet the minimum 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
- Special characters, including exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)

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 **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**.

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

4.3. Getting started

4.3.1. 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.

Context

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**.
5. 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
Area	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.
	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.  Notice ◦ 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, 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 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.

Section	Parameter	Required	Description
Configurations	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.
	Encryption Method	No	The encryption algorithm that is used to encrypt the disk. This parameter is required if you set the Encryption parameter to Yes. Valid values: ◦ AES256 ◦ SM4: stands for ShangMi 4.
	Encryption Key	No	The key that is used to encrypt the disk. This parameter is required if you set the Encryption parameter to Yes.  Note If no key is available, you must create a key in Key Management Service (KMS).
	Use Snapshot	No	Specifies whether to create the disk based on a snapshot. If you select Yes, 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's next

You need to attach the disk to an instance. For more information, see [Attach disks](#).

4.3.2. 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.

Context

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.
 - Click **OK**.

What's 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](#).

4.3.3. Partition and format a disk

4.3.3.1. 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*.
2. Run the `fdisk -l` 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 -l
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         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 bytes
Disk identifier: 0x00000000
```

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 **wq** 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 -l** command to view the partitions.

If **/dev/vdb1** is displayed in the command output, a partition named **vdb1** is created.

```
[root@iz*****eZ ~]# fdisk -l
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          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 bytes
Disk identifier: 0x01ac58fe
Device Boot      Start          End      Blocks   Id  System
/dev/vdb1            1         41610    20971408+   83  Linux
```

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*****eZ ~]# 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
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
/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 ~]# mount -a
[root@iz*****eZ ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/vda1       40G   5.6G   32G   15% /
tmpfs           499M    0   499M    0% /dev/shm
/dev/vdb1       20G   173M   19G    1% /mnt
```

4.3.3.2. 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**.
5. 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.

4.4. Disks

4.4.1. 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  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](#).

4.4.2. 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

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 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**.

4.4.3. 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**.

4.4.4. 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.
 - If the disk is a Shared Block Storage device, the device must be in the **Available** state.

Context

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

- Log on to the ECS console.
 - [Log on to the Apsara Uni-manager Management Console.](#)
 - In the top navigation bar, choose **Products > Elastic Computing > Elastic Compute Service**.
- In the left-side navigation pane, choose **Storage & Snapshots > Disks**.
- In the top navigation bar, select an organization, a resource set, and a region.
- Find the disk that you want to resize, move the pointer over the  icon in the **Actions** column, and then select **Resize Disk**.
- 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.

 **Notice** The new capacity must be greater than the current capacity.

- 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.

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

4.4.5.1. 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.

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 `lsblk` command to check the device names and partition names of the disks, as shown in the following figure.

```
[root@ecs ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
nvme0n1      259:0    0   40G  0 disk
└─nvme0n1p1  259:1    0   40G  0 part /
nvme1n1      259:2    0  100G  0 disk
```

Description of the command output:

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.

4.4.5.2. 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 consist 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 [NVMe 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 list provides instance families that support this 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, see Resize 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
Area	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 resides.
	Zone	Yes	The zone in which the disk resides.
Basic Configurations	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.  Notice ◦ 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 .
	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.

Section	Parameter	Required	Description
	Multi-attach	No	Specifies whether to enable the multi-attach feature for the disk. In this topic, Yes is selected.
	Encryption	No	Specifies whether to encrypt the disk. Default value: No.
	Encryption Method	No	The encryption algorithm that is used to encrypt the disk. This parameter is required if you set the Encryption parameter to Yes . Valid values: ◦ AES256: uses the AES-256 algorithm. ◦ SM4: uses the SM4 algorithm.
	Encryption Key	No	The key that is used to encrypt the disk. This parameter is required if you set the Encryption parameter to Yes .  Note If no key is available, you must create a key in Key Management Service (KMS).
	Use Snapshot	No	Specifies whether to create the disk based on a snapshot. If you select Yes , 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.

 **Note** For more information, see [Create a disk](#).

6. Confirm the configurations and click **Submit**.
7. 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](#).
1. 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**.

 **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.

4.4.6. Encrypt a disk

4.4.6.1. 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.

Context

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.

Parameter	Description
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** drop-down 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.

4.4.6.2. 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**.
5. 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, see [Create a disk](#).

6. Click **Submit**.

4.4.7. 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's 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](#).

4.4.8. Re-initialize disks

4.4.8.1. 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](#).

Context

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's 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.

 **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](#).

4.4.8.2. 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 (`#`) to the beginning of the command line.

```
# /dev/vdb1 /InitTest ext3 defaults 0 0
```

 **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 `:wq` to save the file and exit.

Context

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:
 - 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.
- 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's 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.

4.4.9. 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.

Context

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.

```
[root@ecs ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        441M    0  441M   0% /dev
tmpfs           459M    0  459M   0% /dev/shm
tmpfs           459M  468K  459M   1% /run
tmpfs           459M    0  459M   0% /sys/fs/cgroup
/dev/vda1       40G   2.6G   35G   7% /
/dev/vdb1       40G   49M   38G   1% /mnt
tmpfs           92M    0   92M   0% /run/user/0
```

3. 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.

```
[root@ecs ~]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Mon Sep 14 07:14:22 2020
#
# 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=9f2d3e15-...-0165b4b67864 / ext4 defaults 1 1
UUID=430d44fe-...-5d01bc597839 /mnt ext4 defaults 0 0
[root@ecs ~]#
```

6. Delete the automatic mount configuration information of the file systems in the data disk partitions from `/etc/fstab`.

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.

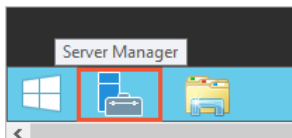
```
#
# /etc/fstab
# Created by anaconda on Mon Sep 14 07:14:22 2020
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=9f2d3e15-...-0165b4b67864 / ext4 defaults 1 1
# UUID=430d44fe-...-5d01bc597839 /mnt ext4 defaults 0 0
```

- iv. Press the ESC key, enter `:wq`, and then press the ENTER key to save your edits and exit the edit mode.

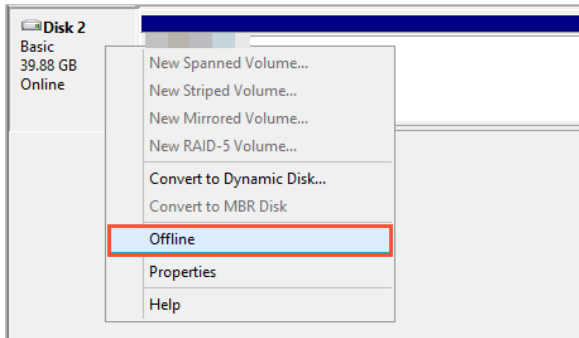
Procedure for a Windows instance

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**.



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's 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](#).

4.4.10. 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**.

4.5. Snapshots

4.5.1. Disk snapshots

4.5.1.1. 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.

Context

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](#).



Notice 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.



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
Snapshot Name	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 (-).  Note The name cannot start with auto because snapshots whose names start with auto are recognized as automatic snapshots.
Snapshot Description	The description of the snapshot. The description must be 2 to 256 characters in length and cannot start with 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.

4.5.1.2. 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  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.

 **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.

Field	Description
Tag	Select a tag key or a tag value to search for the snapshots to which the tag is added.

4.5.1.3. 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's 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](#).

4.5.1.4. 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](#).

Context

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**.

4.5.1.5. 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**.

4.5.2. Snapshot-consistent groups

4.5.2.1. 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.

Context

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 multi-application 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 [Delete one or more snapshots at a time](#).

- 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 <code>http://</code> or <code>https://</code>. ◦ 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 <code>http://</code> or <code>https://</code> .
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's 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](#).

4.5.2.2. 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's 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.

4.5.2.3. 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](#).

 **Notice** 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.

4.5.2.4. Delete a snapshot-consistent group

You can delete a snapshot-consistent group if it is no longer used.

Context

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**.

4.6. Automatic snapshot policies

4.6.1. 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.

Context

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

Parameter	Required	Description
Creation Time	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.  Note The default time zone for the snapshot policy is UTC+8. You can change the time zone based on your business requirements. 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's 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](#).

4.6.2. 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.
 - Click **Advanced Filter**. Specify values for different fields as required and click **Search**. The automatic snapshot policies 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
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.

4.6.3. 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.
5. 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](#).

 **Note** Changes made to the retention period do not affect the existing snapshots and take effect only on subsequent snapshots.

6. Click **OK**.

4.6.4. 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.

Context

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.

4.6.5. 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**.

4.7. Async replication

4.7.1. 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.

Context

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 <code>http://</code> or <code>https://</code>. ◦ 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.

Parameter	Description
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's 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](#).

4.7.2. View replication pairs

This topic describes how to view the created replication pairs.

Procedure

- Log on to the EBS console.
 - [Log on to the Apsara Uni-manager Management Console](#).
 - In the top navigation bar, choose **Products > Elastic Block Storage**.
- In the left-side navigation pane, choose **Enterprise Feature > EBS Async Replication**.
- In the top navigation bar, select an organization and a region.
- 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  icon. The replication pairs with the specified IDs are displayed in the replication pair list.
- 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**.

 **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.

4.7.3. 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.

Context

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 > 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 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.
6. 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.

4.7.4. 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](#).

Context

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 > 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 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.

 **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.
 - o If you click **Copy Data**, the system immediately starts data synchronization.
 - o 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.

4.7.5. 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.

Context

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 > 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 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**.

4.7.6. Use the async replication feature to implement disaster recovery for a primary disk

After you create and activate a replication pair, if the primary disk in the replication pair fails, you can use the async replication feature to implement disaster recovery for the primary disk. This topic describes how to use the async replication feature to implement disaster recovery for a primary disk.

Context

The async 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 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 > 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 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  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 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.

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 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 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. 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**.

 **Notice** 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 Disk C in Zone B of Region A becomes the secondary disk.

6. In the **Operation** 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 that corresponds to 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 **Operation** column to perform a **failover** and **reverse replication** again.
 - iii. After the replication relationship in the replication pair is reverted, click **Activate** in the **Operation** column to activate the replication pair again.

4.7.7. 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.

Context

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 > 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 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.

4.8. Storage sets

4.8.1. 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.

Context

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 > 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 <code>http://</code> or <code>https://</code>. ◦ 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's 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](#).

4.8.2. 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 > 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, select a filter condition, enter the corresponding information in the search box, and then click the  icon. 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.

4.8.3. 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 > 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, 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**.

4.8.4. Delete a storage set

This topic describes how to delete a storage set that is no longer needed.

Context

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 > 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, find the storage set that you want to delete and click **Delete** in the Operation column.
5. In the message that appears, click **OK**.

4.9. Replication pair-consistent groups

4.9.1. 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.

Context

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 > 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 <code>http://</code> or <code>https://</code> . 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's 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](#).

4.9.2. 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 Feature > Replication Pair-Consistent Group**.
3. In the top navigation bar, select an organization and a region.

4. 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  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.

4.9.3. 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.

Context

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 > 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 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](#).

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.

4.9.4. 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](#).

Context

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 > 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 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**.

 **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's 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](#).

4.9.5. 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.

Context

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 > 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 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**.

4.9.6. 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.

Context

- 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 > 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 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.

 **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.

4.9.7. 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.

Context

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 > 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 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.

 **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**.

4.9.8. 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 pair-consistent 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.

Context

Replication pair-consistent groups support the failover and reverse replication sub-features. When the primary disks fail, you can use the failover sub-feature 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 > 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 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.

5. After you create snapshots for the primary disks, click **OK**.

The state of the replication pair-consistent group changes to **Stopped**.

6. 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:

- i. 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.
- ii. Find the replication pair-consistent group on which you have performed a reverse replication. Move the pointer over the  icon in the Operation 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.

4.9.9. Delete a replication pair-consistent group

You can delete a replication pair-consistent group that you no longer need.

Context

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-consistent group](#).

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 > 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.

4.10. Appendix

4.10.1. 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.

Context

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 `aliyun` 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.

 **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 **Close**.
- 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 **Close**.

The removed tags are no longer displayed in the **Tag** drop-down list.

5. Log Service

5.1. 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 log data. Log Service allows you to collect, query, analyze, visualize, consume, and ship log 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 various types of logs in real time, such as events, binary logs, and text logs by using multiple methods. For example, you can use Logtail and JavaScript to collect logs.
- **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.
- **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.

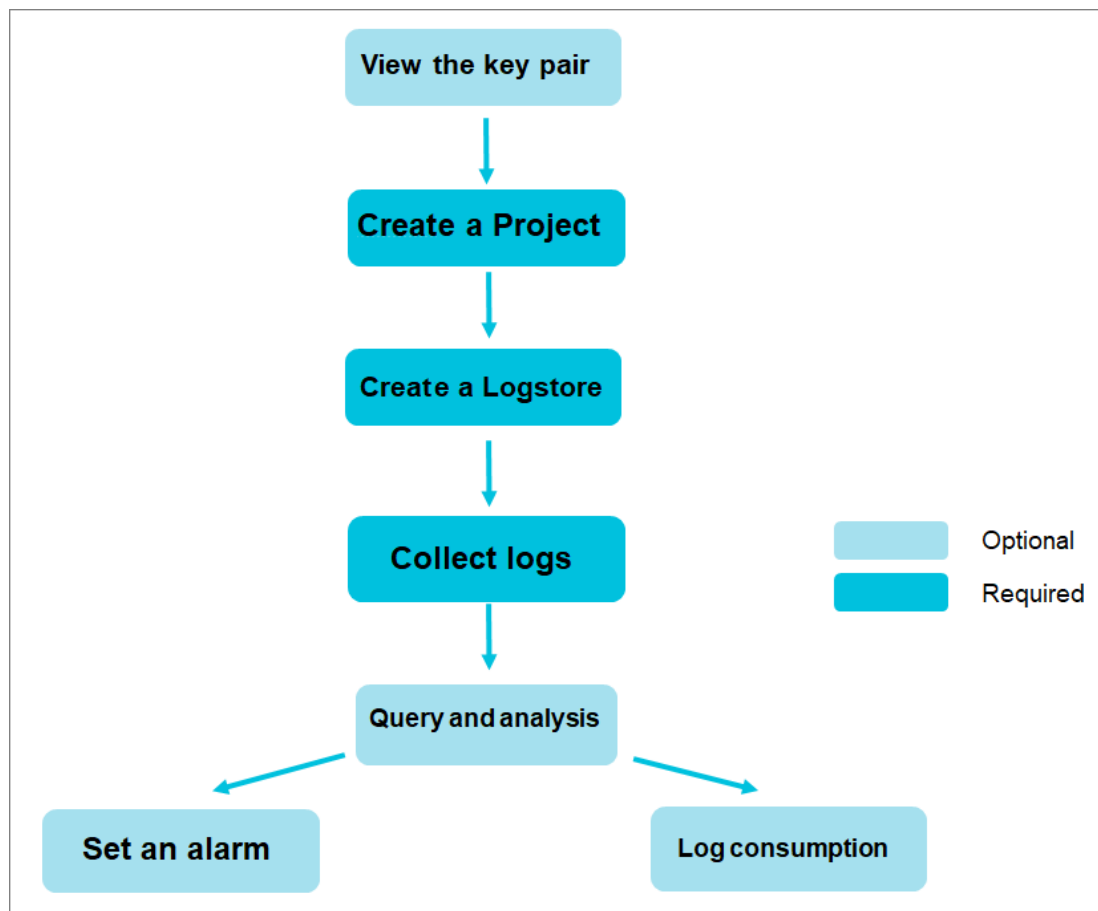
5.2. Quick start

5.2.1. 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.

Procedure



- 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.

- Create a project.

Create a project in a specified region. For more information, see [Create a project](#).

- Create a Logstore.

Create a Logstore for the project and specify the number of shards. For more information, see [Create a Logstore](#).

- Collect text logs.

Select a method to collect log data based on your business requirements. For more information, see [Collect text logs](#).

- 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 [Enable the index feature and configure indexes for a Logstore](#).
- 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 [Query and analysis](#).
- After you query and analyze log data, you can configure charts and dashboards to display query and analysis results. For more information, see [Overview](#) and [Dashboard overview](#).

- 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](#).

5.2.2. Log on to the Log Service console

This topic describes how to log on to the Log Service console.

Prerequisites

- Before you log on to the Apsara Uni-manager Management Console, the endpoint of the console is obtained from the deployment staff.
- 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.
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 The first time that you log on to the Apsara Uni-manager Management Console, you must change the password of your account. For security purposes, your password must meet the minimum 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
- Special characters, including exclamation points (!), at signs (@), number signs (#), dollar signs (\$), and percent signs (%)

3. Click **Log On**.
4. If your account has multi-factor authentication (MFA) enabled, perform 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 **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 more information, see the *Bind a virtual MFA device to enable MFA* topic in *Apsara Uni-manager Management Console User Guide*.

5. In the top navigation bar, choose **Products > Log Service**.
6. 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.

5.2.3. Obtain an AccessKey pair

An AccessKey pair consists of an AccessKey ID and an AccessKey secret. The AccessKey pair is used to implement symmetric encryption to verify the identity of the requester. The AccessKey ID is used to identify a user. The AccessKey secret is used to encrypt the signature string. This topic describes how to obtain an AccessKey pair.

Prerequisites

Only the operation administrators or level-1 organization administrators can obtain the AccessKey pair of an organization.

Context

To call Apsara Uni-manager and cloud service APIs, we recommend that you use the AccessKey pair of a personal account. If you use the AccessKey pair of a personal account, you must configure header parameters as described in the following table for access control.

Parameter	Description
x-acs-regionid	The region ID, such as cn-hangzhou-*.
x-acs-organizationid	The ID of the organization in the Apsara Uni-manager Management Console.
x-acs-resourcegroupid	The ID of the resource set in the Apsara Uni-manager Management Console.
x-acs-instanceid	The ID of the instance on which you want to perform operations.

 **Warning** The AccessKey pairs of personal accounts are under control of the Apsara Uni-manager permission system. AccessKey pairs of organization accounts have higher permissions. For security purposes, organization operations must be approved by administrators.

Obtain the AccessKey pair of a personal account

To obtain the AccessKey pair of a personal account, perform the following operations:

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 **Personal Information**.
3. In the **Apsara Stack AccessKey Pair** section, view your AccessKey pair.

Apsara Stack AccessKey Pair You must use the AccessKey pair when you access Apsara Stack resources.		
The AccessKey pair including the AccessKey ID and AccessKey secret is the credential for you to use Apsara Stack resources with full permissions. You must keep the AccessKey pair confidential.		
Region	AccessKey ID	AccessKey Secret
cn-*****d01	*****	Show

 **Note** The AccessKey pair consists of an AccessKey ID and an AccessKey secret. AccessKey pairs allow you to access Apsara Stack resources with full permissions for your account. You must keep your AccessKey pair confidential.

Obtain the AccessKey pair of an organization

To obtain the AccessKey pair of an organization, perform the following operations:

1. Log on to the Apsara Uni-manager Management Console as an administrator.

2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Organizations**.
4. In the organization navigation tree, click an organization name.
5. In the Current Organization section, click **Management Accesskey**.
6. In the Management AccessKey, view the AccessKey pair of the organization.

 **Note** An AccessKey pair is automatically allocated to each level-1 organization. Subordinate organizations use the same AccessKey pair of their level-1 organization.

5.2.4. 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

 **Note** You can create up to 50 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 parameters and click **OK**. The following table describes the parameters.

Parameter	Description
Project Name	The name of the project. The name must be unique in a region. You cannot change the name after you create the 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 use the internal network of Alibaba Cloud to accelerate log collection.
Service Logs	Select the service logs that you want to store. After you select the service logs, the logs that are generated in the project are stored in a specified project. For more information, see <i>Monitor Log Service</i> in <i>Log Service Developer Guide</i> . ◦ If you select Detailed Logs, operation logs are stored in the specified project. You are billed by using the pay-as-you-go method. ◦ If you select Important Logs, logs of the consumer group latency and Logtail heartbeats are stored in the specified project. This feature is provided free of charge.

Parameter	Description
Log Storage Location	If you select the service logs that you want to store, you must configure this parameter. Valid values: - Automatic creation (recommended) - Current Project - Other projects in the same region as your project

View the endpoint of a project

After you create a project, you can view the endpoint of the project on the **Overview** page.

1. In the Projects section, click the name of the project.
2. On the **Overview** page, view the endpoint of the project.

To access the projects that reside in different regions, you must use different endpoints. To access the projects that reside in the same region over a private network or the Internet, you must use different endpoints.

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: GetProject - Query all projects: ListProject
Modify a project	UpdateProject

For more information, see *API Reference* in *Log Service Developer Guide*.

5.2.5. Manage a Metricstore

This topic describes how to create, modify, and delete a Metricstore in the Log Service console.

Prerequisites

A project is created. For more information, see [Manage a project](#).

Create a 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 + 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.
Permanent Storage	If you turn on Permanent Storage, Log Service permanently stores the collected metric data.  Note You can use an SDK to query the data retention period. The value 3650 indicates that metric data is permanently stored in the Metricstore.
Data Retention Period	The retention period of the collected metric data in the Metricstore. Valid values: 15 to 3000. Unit: days. Metrics whose retention period exceeds the value of this parameter are automatically deleted. If you do not turn on Permanent Storage, you must configure the Data Retention Period parameter.  Note If you shorten the data retention period, Log Service deletes data based on the new retention period within 1 hour. The volume of data that is displayed on the Storage Size(Log) card 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 metric data of the previous four days within 1 hour after the change.
Shards	The number of shards. Log Service provides shards to read and write data. Each shard supports a write speed of 5 MB/s, 500 write operations per second, a read speed of 10 MB/s, and 100 read operations per second. You can create up to 10 shards for each Metricstore. You can create up to 200 shards for each project. For more information, see <i>Log Service Product Introduction</i> .

Modify the configurations of a Metricstore

1. Choose **Time Series Storage > Metricstore**. On the Metricstore tab, find the Metricstore whose configurations you want to modify and choose  > **Modify**.
2. On the **Metricstore Attribute** page, click **Modify**.
 - Change the data retention period. For more information, see *Create a Metricstore*.
 - Manage shards.

When you create a Metricstore, the Shards parameter is set to 2 by default. You can split or merge the shards based on your business requirements. For more information, see [Manage shards](#).
3. Click **Save**.

Delete a Metricstore

 Notice

- 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 you are not authorized to delete a Metricstore by using your Alibaba Cloud account, submit a ticket.

1. Choose **Time Series Storage > Metricstore**. On the Metricstore tab, find the Metricstore that you want to delete and choose  > **Delete**.

 **Warning** After you delete a Metricstore, all metric data in the Metricstore is deleted and cannot be restored. Proceed with caution.

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

5.2.6. 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

 **Note** You 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 Plus 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 and send data from HTML, HTML5, iOS, or Android platforms to Log Service by using the web tracking feature.
Permanent Storage	If you turn on Permanent Storage, Log Service permanently stores the collected logs.  Note You can call an API operation to query the data retention period. If the value is 3650, the logs are permanently stored.

Parameter	Description
Data Retention Period	If you turn off Permanent Storage, you must configure this parameter to specify 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.  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 speed of 5 MB/s, 500 write operations per second, a read speed of 10 MB/s, and 100 read operations per second. You can create up to 10 shards in each Logstore. You can create up to 200 shards in each project. For more information, see <i>Log Service Product Introduction</i> .
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.
Maximum Shards	If you turn on Automatic Sharding , you must configure this parameter to specify the maximum number of shards that can be created. 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 00:00:00 Thursday, January 1, 1970.

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. Click **Save**.

Delete a Logstore

If you no longer need a Logstore, you can delete the Logstore in the Log Service console.

Notice

- 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 the Logstore, we recommend that you stop writing data to the Logstore and make sure that all data in the Logstore is shipped before you delete the Logstore.
- If you are not authorized to delete a Logstore by using your Alibaba Cloud 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**.

 **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: GetLogstore • Query all Logstores: ListLogstore
Modify a Logstore	UpdateLogstore

5.2.7. 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.

Context

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.

 **Note** You can split only a shard that is in the readwrite state.

Total Shards:2 (Read/Write Instances:2 , Read-only Instances:0)				
Id ↓↑	Status ↓↑ ⚙	Beginkey/EndKey	Created At ↓↑	Actions
0	readwrite	00000000000000000000000000000000 80000000000000000000000000000000	2020-02-24 17:50:37	Split Merge
1	readwrite	80000000000000000000000000000000 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	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)				
Id ↓↑	Status ↓↑ ⚙	Beginkey/EndKey	Created At ↓↑	Actions
0	readwrite	00000000000000000000000000000000 80000000000000000000000000000000	2020-02-24 17:50:37	Split Merge
1	readwrite	80000000000000000000000000000000 00000000000000000000000000000000	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

5.2.8. Terms

5.2.8.1. 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.

Term	Description
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 MD5 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 MD5 hash range [00000000000000000000000000000000,fffffffffffffffffffffffffffffffff).
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 <i>Obtain the endpoint of Log Service</i> 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 <i>Obtain an AccessKey pair</i> in Developer Guide.

Data collection

Term	Description
Logtail	Logtail is used by Log Service to collect logs. For more information, see <i>Collection by Logtail</i> 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 <i>Collection by Logtail</i> 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 <i>Logtail machine group</i> 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 <i>Log search overview</i> in the "Query and analysis" section.

Term	Description
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, see <i>Log analysis overview</i> 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 <i>Query and analysis</i> in the "Time series storage" section.
query statement	A query statement is in the <code>Search statement Analytic statement</code> 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 <i>Query and analysis</i>.
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 <i>Configure indexes</i> 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 <i>Alerts</i>.

5.2.8.2. 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
Topic	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 <code>access_log</code> and <code>operation_log</code> , for website logs based on log types. For more information, see <i>Topic</i> .	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: <code>__time__</code> <code>__source__</code> <code>__topic__</code> <code>__partition_time__</code> <code>__extract_others__</code> <code>__extract_others__</code> - 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 <code>PutLogs</code> operation to write logs to a specified Logstore. - System tags: the tags that are added by Log Service. The tags include <code>__client_ip__</code> and <code>__receive_time__</code>.	The field value is in the dictionary format. The keys and the values are strings. The field name is prefixed by <code>__tag__</code> .

Example

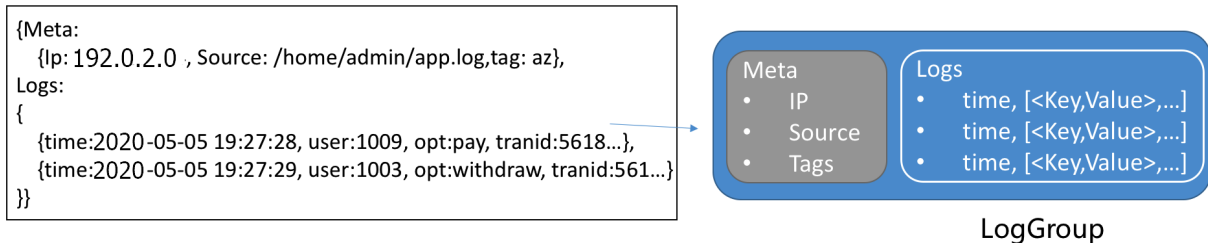
The following sample website access log shows the mapping between the raw log and the data model supported by Log Service.

```
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"
```

5.2.8.3. 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.



5.2.8.4. 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*.

5.2.8.5. 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 query and analysis operations.

5.2.8.6. 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 query and analyze metrics.

5.2.8.7. 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-Z0-9_]*` 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.

Key	Description	Example
<code>__name__</code>	The name of the metric.	<code>nginx_ingress_controller_response_size</code>
<code>__labels__</code>	The labels of the metric. Format: <code>{key}##{value}{{key}##{value}}{{key}##{value}}</code> .  Note The labels must be sorted by key in alphabetical order.	<code>app##ingress-nginx controller_class##nginx controller_namespace##kube-system controller_pod##nginx-ingress-controller-589877c6b7-hw9cj</code>
<code>__time_nano__</code>	The timestamp of a sample. The value is accurate to the nanosecond.	<code>1585727297293000000</code>
<code>__value__</code>	The value of a sample.	<code>36.0</code>

5.2.8.8. 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

Each shard has an MD5 value range, and each range is a left-closed, right-open interval in the [BeginKey, EndKey) format. Each range does not overlap the ranges of other shards. The entire MD5 value range is within the following range: [00000000000000000000000000000000,ffffffffffffffffffffffffffffffff). You can specify the number of shards when you create a Logstore or Metricstore. Log Service evenly divides the entire MD5 value range for each shard.

- **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.

MD5 value range

Shard ID	Value range
Shard0	[00000000000000000000000000000000,40000000000000000000000000000000)
Shard1	[40000000000000000000000000000000,80000000000000000000000000000000)
Shard2	[80000000000000000000000000000000,c0000000000000000000000000000000)
Shard3	[c0000000000000000000000000000000,ffffffffffffffffffffffffffffffff)

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.

 Notice

- 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 readonly 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.

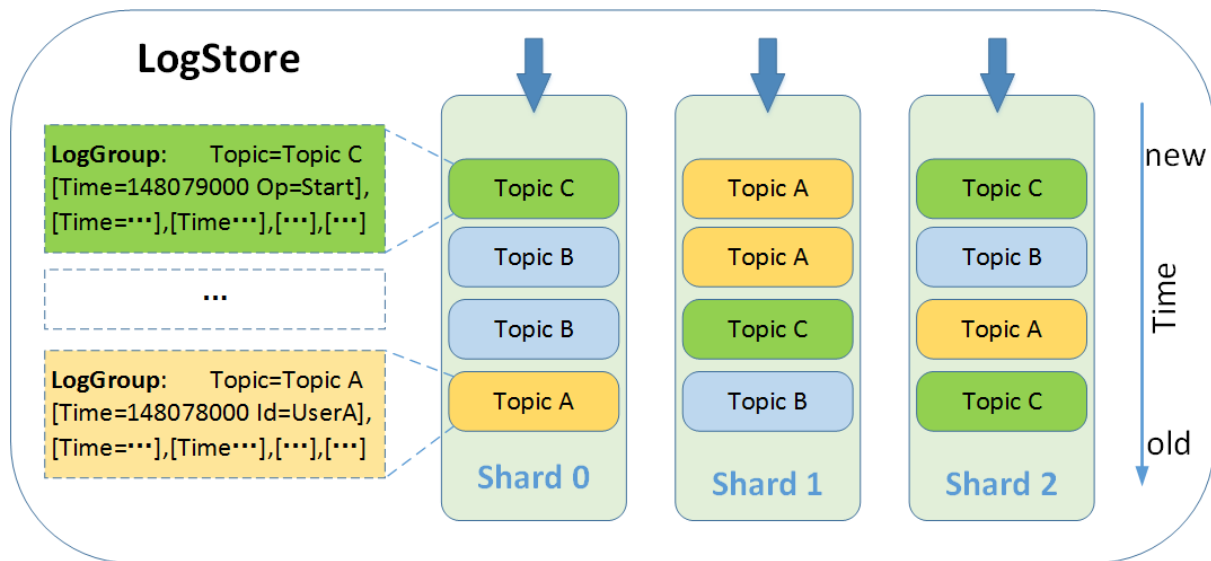
5.2.8.9. Topic

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.



5.3. Data collection

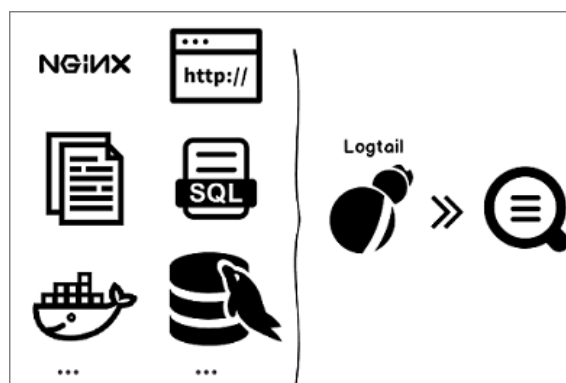
5.3.1. Collection by Logtail

5.3.1.1. Overview

5.3.1.1.1. 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.

Logtail-based log collection



Benefits

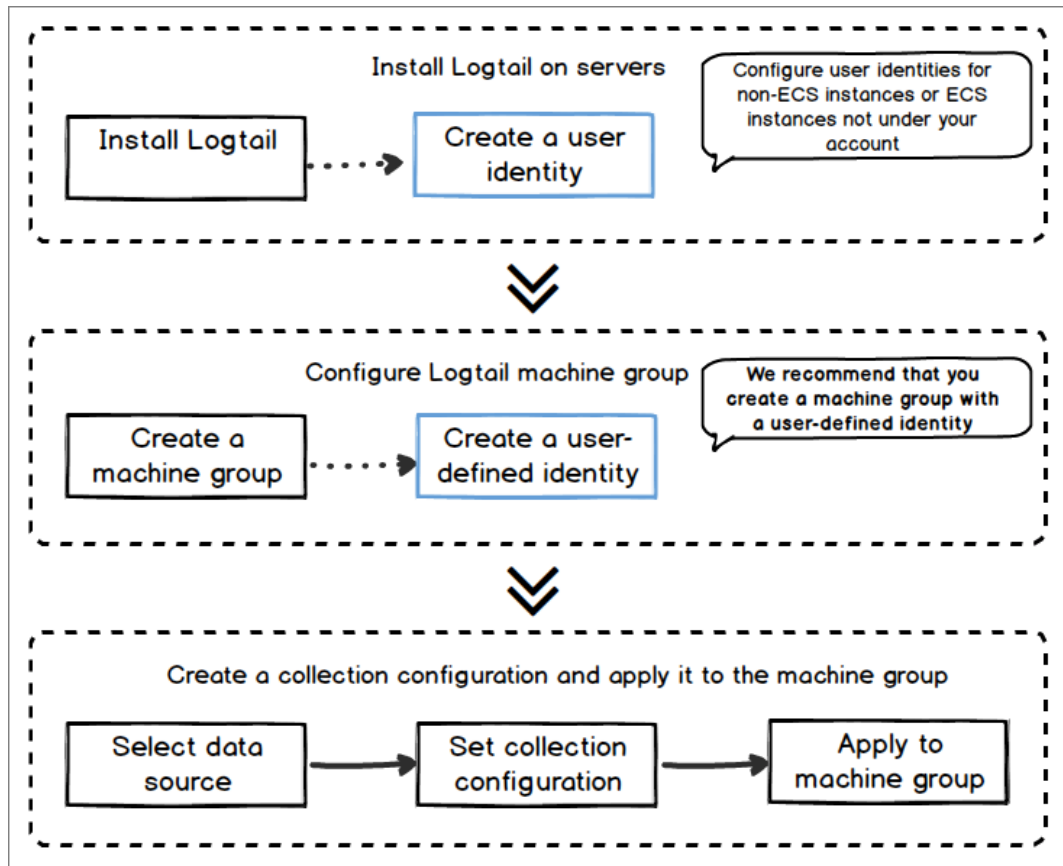
- Supports non-intrusive log collection based on log files. You do not need to modify your application code. Your applications are not affected when Logtail collects logs.
- Allows you to collect text logs, binary logs, HTTP data, and container logs.
- Allows you to collect logs from standard containers, 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 in Linux](#) and [Install Logtail in Windows](#).

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 a machine group based on a server IP address](#) and [Create a machine group based on a custom ID](#).

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 in Linux](#) or [Install Logtail in Windows](#).
 - 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 <code>app.LOG</code> file is renamed <code>app.LOG.1</code> and <code>app.LOG.2</code> during log rotation. You can specify the file to which collected logs are written, for example, <code>app.LOG</code>. 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.

Feature	Description
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.  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 start up parameters that are specified in the start up 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.

5.3.1.1.2. 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 logs](#).

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.

 **Notice** 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.

- If you turn on **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 [Set Logtail startup parameters](#).

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.

5.3.1.1.3. 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 [Set Logtail startup parameters](#).

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 32-bit 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/ilogtail_config.json` command, the following output is returned:

```
{
  "config_server_address" : "http://logtail.cn-hangzhou-intranet.log.aliyuncs.com",
  "data_server_list" :
  [
    {
      "cluster" : "cn-hangzhou",
      "endpoint" : "cn-hangzhou-intranet.log.aliyuncs.com"
    }
  ],
  "cpu_usage_limit" : 0.4,
  "mem_usage_limit" : 100,
  "max_bytes_per_sec" : 2097152,
  "process_thread_count" : 1,
  "send_request_concurrency" : 4,
  "streamlog_open" : false
}
```

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 an account ID on a server](#).

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 machine group based on a custom ID](#).

Note When you create a custom ID-based machine group, you must configure the `user_defined_id` file.

• File path

- 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
```

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 in Log Service, 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

If you run the `cat /usr/local/ilogtail/user_log_config.json` command, the following output is returned:

```
{
  "metrics" : {
    "##1.0##k8s-log-cl2ba2028*****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"
              },
              "IncludeLabel" : {
                "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-cl2ba2028c*****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`.
- 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" : "E5F93BC6-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. 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 addresses 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/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/ilogtail/conf/ap-southeast-2/ilogtail_config.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://logtail.ap-southeast-2-intranet.log.aliyuncs.com",
  "data_server_list" : [
    {
      "cluster" : "ap-southeast-2",
      "endpoint" : "ap-southeast-2-intranet.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##sfs-zc-test-hz-pub$docker-std
out-config,k8s-stdout] open file for read, file:/logtail_host/var/lib/docker/containers/7f46afec
6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f114
8b2e2f31bd3410f5b2d624-json.log offset:40379573 status:794354-64769-40379963
2018-09-13 03:03:26 [INF] [log_file_reader.go:221] [ReadOpen] [##1.0##k8s-log-cl2ba2028cfb444238cd9
ac1286939f0b$docker-stdout-config,k8s-stdout] open file for read, file:/logtail_host/var/lib/doc
ker/containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59
ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log offset:40379573 status:794354-64769-40
379963
2018-09-13 03:04:26 [INF] [log_file_reader.go:308] [CloseFile] [##1.0##sfs-zc-test-hz-pub$docker-st
dout-config,k8s-stdout] close file, reason:no read timeout file:/logtail_host/var/lib/docker/
containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9c
dfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log offset:40379963 status:794354-64769-403799
63
2018-09-13 03:04:27 [INF] [log_file_reader.go:308] [CloseFile] [##1.0##k8s-log-cl2ba2028cfb444238cd
9ac1286939f0b$docker-stdout-config,k8s-stdout] close file, reason:no read timeout file:/logta
il_host/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/
7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log offset:40379963 sta
tus:794354-64769-40379963
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/overlay2/947db346695a1f65e63e582ecfd10ae1f57019alb99260b6c83d00fcd1892874/diff/var/log\",\n  \"Tags\" : [\n    \"nginx-type\",\n    \"access-log\",\n    \"_image_name_\",\n    \"registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest\",\n    \"_container_name_\",\n    \"nginx-log-demo\",\n    \"_pod_name_\",\n    \"nginx-log-demo-h21zc\",\n    \"_namespace_\",\n    \"default\",\n    \"_pod_uid_\",\n    \"87e56ac3-b65b-11e8-b172-00163f008685\",\n    \"_container_ip_\",\n    \"172.20.4.224\",\n    \"purpose\",\n    \"test\"\n  ]\n}"
    }
  ],
  "version" : "0.1.0"
}
```

5.3.1.2. Installation

5.3.1.2.1. Install Logtail on a Linux server

This topic describes how to install, upgrade, 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
- Glibc 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 starts when the system reboots.

1. Run the following command to download the Logtail installer:

```
wget http://${service:sls-backend-server:sls_data.endpoint}/logtail.sh -O 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 Overview page of a Logstore.

2. Run the installation command.

Start Linux PowerShell and run the following command as an administrator to install Logtail:

```
./logtail.sh install
```

3. Configure a user identifier.

View the version of Logtail

Go to the installation directory and open the `/usr/local/ilogtail/app_info.json` file. The value of the `logtail_version` field indicates the version of Logtail. Run the `cat /usr/local/ilogtail/app_info.json` command to view the version of Logtail. The following example shows a response:

```
{
  "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.ali1113.el5.x86_64; #1 SMP Thu Jul 4 20:09:15 CST 2013; x86_64",
  "update_time" : "2018-01-05 13:19:08"
}
```

Upgrade Logtail

You can use the Logtail installation script `logtail.sh` to upgrade Logtail. The installation script selects an upgrade method based on the configurations of the Logtail that is installed.

 **Note** During the upgrade, Logtail is temporarily stopped, and all files, except the configuration files and checkpoint files, are overwritten.

Run the following command to upgrade Logtail:

```
# Download the Logtail installer.
wget http://${service:sls-backend-server:sls_data.endpoint}/logtail.sh -O logtail.sh;
chmod 755 logtail.sh
# Upgrade Logtail.
sudo ./logtail.sh upgrade
```

Check the upgrade result:

```
# The upgrade is successful.
Stop logtail successfully.
ilogtail is running
Upgrade logtail success
{
  "UUID" : "****",
  "hostname" : "****",
  "instance_id" : "****",
  "ip" : "****",
  "logtail_version" : "0.16.11",
  "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" : "2018-08-29 15:01:36"
}
# The upgrade fails because the current version is the latest version.
[Error]: Already up to date.
```

Start and stop Logtail

- Start Logtail

Run the following command as an administrator to start Logtail:

```
/etc/init.d/ilogtaild start
```

- Stop Logtail

Run the following command as an administrator to stop Logtail:

```
/etc/init.d/ilogtaild stop
```

Uninstall Logtail

Run the following command as an administrator to uninstall Logtail in Linux PowerShell:

```
wget http://${service:sls-backend-server:sls_data.endpoint}/logtail.sh -O logtail.sh
chmod 755 logtail.sh
./logtail.sh uninstall
```

5.3.1.2.2. Install Logtail in Windows

This topic describes how to install Logtail on a Windows server.

Supported systems

Logtail supports the following Windows operating systems:

- Windows 7 (Client) 32-bit
- Windows 7 (Client) 64-bit
- Windows Server 2008 32-bit
- Windows Server 2008 64-bit
- Windows Server 2012 64-bit
- Windows Server 2016 64-bit

Procedure

1. Download the installation package.

Run the following command to download the installation package:

```
wget http://${service:sls-backend-server:sls_data.endpoint}/windows/logtail_installer.zip
```

 **Note** You must replace `${service:sls-backend-server:sls_data.endpoint}` in the command with the actual endpoint. For more information about endpoints, see [View the information of a project](#).

2. Decompress the `logtail_installer.zip` package to the current directory.
3. Run the installation command.

Run Windows PowerShell or Command Prompt as an administrator. Enter the `logtail_installer` directory, and then run the installation command based on the network type.

```
.\logtail_installer.exe install me-east-1
```

 **Note** You must replace `${region}` in the command with the actual endpoint. For more information about endpoints, see [View the information of a project](#).

4. [Configure an account ID for a server](#).

Installation directory

After you run the installation command, Logtail is installed in the specified directory. The directory cannot be changed. In the directory, you can [View the version of Logtail](#) in the `app_info.json` file or [Uninstall Logtail](#).

The installation directory is as follows:

- 32-bit Windows: `C:\Program Files\Alibaba\Logtail`
- 64-bit Windows: `C:\Program Files (x86)\Alibaba\Logtail`

 **Note** You can run 32-bit or 64-bit applications in a 64-bit Windows operating system. However, the operating system stores 32-bit applications in separate x86 folders to ensure compatibility.

Logtail for Windows is a 32-bit application. Therefore, it is installed in the `Program Files (x86)` folder in 64-bit Windows. If Logtail for 64-bit Windows becomes available in the future, it will be installed in the `Program Files` folder.

View the version of Logtail

To view the version of Logtail, go to the [default installation directory](#), and then use the notepad or another text editor to open the `app_info.json` file. The `logtail_version` field shows the version of Logtail.

In the following example, the version of Logtail is 1.0.0.0:

```
{
  "logtail_version" : "1.0.0.0"
}
```

Upgrade Logtail

- Automatic upgrade

Logtail later than 1.0.0.0 is automatically upgraded in Windows.

- Manual upgrade

Logtail earlier than 1.0.0.0 must be manually upgraded. The manual upgrade procedure is the same as the [installation procedure](#).

 **Note** During a manual upgrade, the files in the original installation directory are deleted. We recommend that you back up the files before you perform a manual upgrade.

Start and stop Logtail

Open the **Control Panel**, choose **System and Security > Administrative Tools**, and then double-click **Services**.

Find the service based on your Logtail version.

- Logtail 0.x.x.x: LogtailWorker.
- Logtail 1.0.0.0 and later: LogtailDaemon.

Perform the following operations as required:

- Start Logtail

Right-click the service and select **Start** from the shortcut menu.

- Stop Logtail

Right-click the service and select **Stop** from the shortcut menu.

- Restart Logtail

Right-click the service and select **Restart** from the shortcut menu.

Uninstall Logtail

Run Windows PowerShell or Command Prompt as an administrator. Enter the `logtail_installer` directory, and then run the following command:

```
.\logtail_installer.exe uninstall
```

After Logtail is uninstalled, the **installation directory** is deleted. However, some residual configuration data is still maintained in the `C:\LogtailData` directory. You can manually delete the data. The residual configuration data includes the following information:

- **checkpoint**: checkpoints of all plug-ins, for example, the Windows event log plug-in.
- **logtail_checkpoint**: checkpoints of Logtail.
- **users**: IDs of Apsara Stack tenant accounts.

5.3.1.2.3. Set the startup parameters of Logtail

Log Service limits the collection performance of Logtail. This way, Logtail does not consume an excessive number of server resources and affect other services. If you want to improve the collection performance of Logtail, modify the startup parameters of Logtail.

Scenarios

You can modify the startup parameters of Logtail in the following scenarios:

- You want to collect a large number of log files and the log files occupy a large amount of memory. For example, you want to collect more than 100 files or monitor more than 5,000 files in specific directories at the same time.
- Log data is transmitted at a high speed, which causes high CPU utilization. For example, Logtail collects log data at a speed higher than 2 MB/s in simple mode and at a speed higher than 1 MB/s in full regex mode.
- Logtail sends data to Log Service at a speed higher than 20 MB/s.

Set the startup parameters of Logtail

1. On the server on which Logtail is installed, open the `/usr/local/ilogtail/ilogtail_config.json` file.
2. Set the startup parameters of Logtail based on your business requirements.

The following example shows how to configure startup parameters:

```
{
  ...
  "cpu_usage_limit" : 0.4,
  "mem_usage_limit" : 100,
  "max_bytes_per_sec" : 2097152,
  "process_thread_count" : 1,
  "send_request_concurrency" : 4,
  "buffer_file_num" : 25,
  "buffer_file_size" : 20971520,
  "buffer_file_path" : "",
  ...
}
```

Notice

- The following table describes the startup parameters that are commonly used for Logtail. You can use the default values for other startup parameters.
- You can add and modify startup parameters based on your business requirements.

The following table describes the startup parameters of Logtail.

Startup parameters of Logtail

Parameter	Type	Description	Example
cpu_usage_limit	double	The CPU utilization threshold for Logtail. The threshold is specified based on a single-core CPU. - Valid values: 0.1 to the number of CPU cores of the current server. - Default value: 2. For example, if you set the parameter to 0.4, Logtail automatically restarts if the CPU utilization remains higher than 40% for 5 minutes. In most cases, a single core can collect logs at a speed of 24 MB/s in simple mode and 12 MB/s in full regex mode.	<pre>"cpu_usage_limit" : 0.4</pre>
mem_usage_limit	int	The memory usage threshold for Logtail. Unit: MB. - Valid values: 128 to the available memory size of the current server. - Default value: 2048. For example, if you set the parameter to 100, the memory usage is limited to 100 MB. Logtail restarts when the memory usage exceeds 100 MB. If you want to collect more than 1,000 log files at the same time, you can increase the value of this parameter.	<pre>"mem_usage_limit" : 1000</pre>
max_bytes_per_sec	int	The maximum speed at which Logtail sends raw data. Unit: bytes per second. - Valid values: 1024 to 52428800. - Default value: 20971520. For example, if you set this parameter to 2097152, the speed is limited to 2 MB/s.  Notice If you set this parameter to a value that is greater than 20971520, the speed is not limited. The value 20971520 indicates that the speed is 20 MB/s.	<pre>"max_bytes_per_sec" : 2097152</pre>
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 use the default value of this parameter.	<pre>"process_thread_count" : 1</pre>

Parameter	Type	Description	Example
send_request_concurrency	int	The number of concurrent requests that are sent by Logtail to asynchronously send data. - Valid values: 1 to 1000. - Default value: 20. If write transactions per second (TPS) is high, you can set this parameter based on your business requirements. Each concurrent request supports a network throughput of 0.5 MB/s to 1 MB/s. The number of concurrent requests varies based on the network latency.  Note If the value of this parameter is excessively large, concurrent requests may occupy an excessive number of network ports. In this case, you must adjust the TCP parameters.	<pre>"send_request_concurrency" : 4</pre>
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 data that needs to be written exceeds the threshold that you specify, Logtail caches parsed logs to the on-premises files in the installation directory. After the network is recovered and data that needs to be written does not exceed the threshold, Logtail retries to send the cached logs.	<pre>"buffer_file_num" : 25</pre>
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 <code>buffer_file_size</code> parameter by the value of the <code>buffer_file_num</code> parameter.	<pre>"buffer_file_size" : 20971520</pre>
buffer_file_path	String	The directory in which you want to store cached files. Default value: null. This value indicates that cached files are stored in the <code>/usr/local/ilogtail</code> directory in which Logtail is installed. If you modify this parameter, you must move the cached files whose names match <code>logtail_buffer_file_*</code> to the directory that is specified by the new value of the parameter. This way, Logtail can read, send, and delete the cached files.	<pre>"buffer_file_path" : ""</pre>

Parameter	Type	Description	Example
bind_interface	String	The name of the Network Interface Card (NIC) that you want to associate with the server. Default value: null. This value indicates that an available NIC is automatically associated with the server. If you set this parameter to eth1, Logtail uses the NIC named eth1 to upload logs.  Note This parameter is available only if Logtail runs on the Linux operating system.	<pre>"bind_interface" : ""</pre>
check_point_filename	String	The path in which the checkpoint file of Logtail is stored. Default value: <code>/tmp/logtail_check_point</code>. We recommend that you modify the path, and then mount the path on the host as a Docker user. If you perform the operations as another type of user, Logtail collects duplicate files when a Docker container is released due to the loss of checkpoint information. For example, if you set the check_point_filename parameter in the Docker container to <code>/data/logtail/check_point.dat</code>, and then add <code>-v /data/docker1/logtail:/data/logtail</code> to the startup command, the <code>/data/docker1/logtail</code> directory of the host is mounted on the <code>/data/logtail</code> directory of the Docker container.	<pre>"check_point_filename" : /tmp/logtail_check_point</pre>
user_config_file_path	String	The path of the Logtail configuration. By default, Logtail configurations are stored in the directory in which the binary process is stored, and the file name is <code>user_log_config.json</code>. We recommend that you modify the path, and then mount the path on the host as a Docker user. If you perform the operations as another type of user, Logtail collects duplicate files when a Docker container is released due to the loss of checkpoint information. For example, if you set the user_config_file_path parameter in the Docker container to <code>/data/logtail/user_log_config.json</code>, and then add <code>-v /data/docker1/logtail:/data/logtail</code> to the startup command, the <code>/data/docker1/logtail</code> directory of the host is mounted to the <code>/data/logtail</code> directory of the Docker container.	<pre>"user_config_file_path" : user_log_config.json</pre>
discard_old_data	Boolean	Specifies whether to drop historical logs. Default value: true. This value indicates that the logs that were generated 12 hours or more before the current time are dropped.	<pre>"discard_old_data" : true</pre>
working_ip	String	The server IP address that is reported by Logtail to Log Service. Default value: null. This value indicates that Log Service automatically obtains the server IP address.	<pre>"working_ip" : ""</pre>
working_hostname	String	The server hostname that is reported by Logtail to Log Service. Default value: null. This value indicates that Log Service automatically obtains the server hostname.	<pre>"working_hostname" : ""</pre>

Parameter	Type	Description	Example
max_read_buffer_size	long	The maximum size of each log. Default value: 524288. Unit: bytes. If the size of a log exceeds 524288 bytes or 512 KB, you can change the value of this parameter.	<pre>"max_read_buffer_size" : 524288</pre>
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 network is unstable or the connection fails to be established, you can change the value of this parameter.	<pre>"oas_connect_timeout" : 5</pre>
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 network is unstable or the connection fails to be established, you can change the value of this parameter.	<pre>"oas_request_timeout" : 10</pre>
data_server_port	long	If you set the data_server_port parameter to true, Logtail transmits data to Log Service by using the HTTPS protocol. This parameter is available only for Logtail V1.0.10 or later.	<pre>"data_server_port": 443</pre>
enable_log_time_auto_adjust	Boolean	If you set the enable_log_time_auto_adjust parameter to true, the log time is automatically changed to the local time of the server. To ensure data security, Log Service checks the time information in requests, such as the requests sent by Logtail. Log Service rejects requests that are sent 15 minutes earlier or 15 minutes later than the time in Log Service. The point in time at which Logtail sends a request is considered as the local time of the server. In some test scenarios, the local time must be set to a future point in time. If you change the local time of the server, Log Service rejects the request and Logtail fails to write data to Log Service. You can synchronize the log time to the local time of the server based on this parameter. This parameter is available only for Logtail V1.0.19 or later.  Notice - If you enable the adaptive feature, the offset between the time in Log Service and the local time of the server are added to the log time. The offset is updated only if Log Service rejects a request. Therefore, the log time in the query result of Log Service may be different from the point in time when the log is written. - Part of the logic of Logtail changes based on the increment of the system time. We recommend that you restart Logtail after you change the local time of the server.	<pre>"enable_log_time_auto_adjust": true</pre>

Parameter	Type	Description	Example
accept_multi_config	Boolean	Specifies whether to allow multiple Logtail clients to collect data from the same file. Default value: false. This value indicates that only one Logtail client can collect the data of a file. By default, only one Logtail client can collect the data of a file. You can set this parameter to true to allow multiple clients to collect data from the same file. Each Logtail client has an independent collection process. If multiple Logtail clients are allowed to collect data from the same file, the CPU utilization and the memory usage increase. This parameter is available only for Logtail V0.16.26 or later.	<pre>"accept_multi_config": true</pre>

- Restart Logtail for the modified configurations 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 relationships between environment variables and the startup parameters of Logtail. For information about the startup parameters of Logtail, see .

Relationships between environment variables and the startup parameters of Logtail

Parameter	Environment variable	Priority	Supported versions
cpu_usage_limit	cpu_usage_limit	If you use environment variables and configuration files 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		
max_bytes_per_sec	max_bytes_per_sec		
process_thread_count	process_thread_count		
send_request_concurrency	send_request_concurrency		
check_point_filename	ALIYUN_LOGTAIL_CHECK_POINT_PATH	If you use environment variables and configuration files to modify the startup parameters of Logtail, the modifications from the environment variables take effect.	Logtail V0.16.36 or later

5.3.1.3. Logtail machine group

5.3.1.3.1. 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.

 **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.

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 server 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 machine group based on a custom ID](#).

5.3.1.3.2. Configure a user identifier

This topic describes how to specify the ID of an Apsara Stack tenant account as a user identifier on a server.

Prerequisites

- A server is available.

The server can be an Elastic Compute Service (ECS) instance that belongs to another Apsara Stack tenant account, a server that is provided by a third-party cloud service provider, or a self-managed data center.

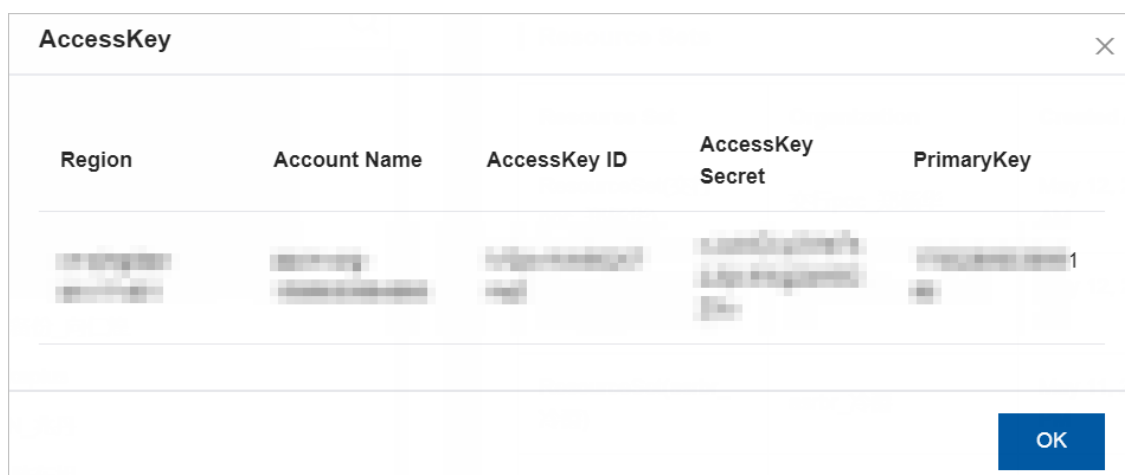
- Logtail is installed on the server from which you want to collect logs. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).

Context

If your server is an ECS instance that belongs to another Apsara Stack tenant account, a server that is provided by a third-party cloud service provider, or a self-managed data center, you must specify the ID of your Apsara Stack tenant account as a user identifier on your server after Logtail is installed. Then, the Apsara Stack tenant account can use Logtail to collect logs from the server. If you do not configure a user identifier on your server, Log Service cannot receive the heartbeat of the server and Logtail cannot collect logs from the server.

Procedure

1. View the ID of your Apsara Stack tenant account.
 - i. Log on to the Log Service console. For more information, see [Log on to the Log Service console](#).
 - ii. In the top navigation bar, click **Enterprise**.
 - iii. In the left-side navigation pane, click **Organizations**.
 - iv. Select the account that you want to view and click **Obtain an accesskey**.
 - v. In the **AccessKey** dialog box, view the ID of the Apsara Stack tenant account.



2. Log on to the server and specify the ID of the Apsara Stack tenant account as a user identifier.

- o Linux server:

In the `/etc/ilogtail/users` directory, create a file and set the name of the file to the ID of the Apsara Stack tenant account. Example:

```
touch /etc/ilogtail/users/1*****
touch /etc/ilogtail/users/1*****
```

- o Windows server:

In the `C:\LogtailData\users` directory, create a file and set the name of the file to the ID of the Apsara Stack tenant account. For example, you can set the file name to `C:\LogtailData\users\1*****`.

Note

- o If the `/etc/ilogtail/users` directory does not exist, you must create the directory.
- o You can configure the IDs of multiple Apsara Stack tenant accounts on the same server.
- o After you configure or delete a user identifier, the change takes effect within 1 minute.
- o If you no longer need a user identity file on a server, we recommend that you delete the file from the server at the earliest opportunity.

5.3.1.3.3. Create an IP address-based machine group

Log Service allows you to create an IP address-based machine group. This topic describes how to create an IP address-based machine group in the Log Service console.

Prerequisites

- A project and a Logstore are created.
- At least one server is available.
 - If you use an Elastic Compute Service (ECS) instance, you must make sure that the ECS instance and Log Service project belong to the same Apsara Stack tenant account and reside in the same region.
 - If the ECS instance belongs to another Apsara Stack tenant account, you must configure a user identifier for the ECS instance before you create an IP address-based machine group. If the server is provided by a third-party cloud service provider or is deployed in a self-managed data center, you must also configure a user identifier. For more information, see [Configure an account ID on a server](#).
- Logtail is installed on the server. For more information, see [Install Logtail in Linux](#) and [Install Logtail in Windows](#).

Procedure

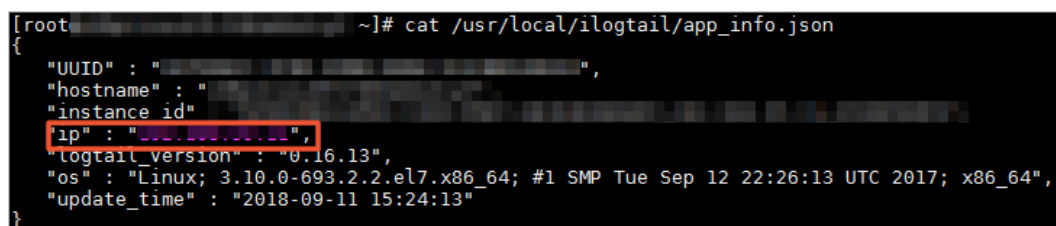
1. Obtain the IP addresses of the servers.

The IP addresses that are obtained by Logtail are recorded in the `ip` field of the `app_info.json` file.

On the servers where Logtail is installed, you can go to the following path to check the `app_info.json` file:

- 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 the IP address of the server in Linux.



```
[root@~]# cat /usr/local/ilogtail/app_info.json
{
  "UUID" : " ",
  "hostname" : " ",
  "instance_id" : " ",
  "ip" : "10.10.10.10",
  "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" : "2018-09-11 15:24:13"
}
```

2. [Log on to the Log Service console](#).
3. In the Projects section, click the project in which you want to create a machine group.
4. In the left-side navigation pane, click the **Machine Groups** icon.
5. Click the  icon next to Machine Groups and select **Create Machine Group**.

You can also create a machine group in the Logtail configuration wizard.

6. In the **Create Machine Group** panel, set the parameters and click **OK**. The following table describes the parameters.

Parameter	Description
Name	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.  Notice After the machine group is created, you cannot modify its name. Proceed with caution.
Identifier	Select IP Addresses.

Parameter	Description
Topic	The topic of the machine group. This topic is used to differentiate log data that is generated on different servers. For more information, see Log topic .
IP address	Enter the IP addresses that are obtained in Step 1. Note - If a machine group contains multiple servers, you must separate the IP addresses with line feeds. - You cannot 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 status of the machine group.

If the **Heartbeat** status is **OK**, the server is connected to Logtail. If the status is **FAIL**, click **Automatic Retry**.

Result

You can view the created machine group in the Machine Groups list.

5.3.1.3.4. Create a custom ID-based machine group

Log Service allows you to create a custom ID-based machine group. This topic describes how to create a custom ID-based machine group in the Log Service console.

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, you must make sure that the ECS instance and Log Service project belong to the same Apsara Stack tenant account and reside in the same region. If the ECS instance belongs to another Apsara Stack tenant account, you must configure a user identifier for the ECS instance before you create a custom ID-based machine group. If the server is provided by a third-party cloud service provider or is deployed in a self-managed data center, you must also configure a user identifier. For more information, see [Configure an account ID on a server](#).
 - Logtail is installed on the server. For more information, see [Install Logtail on ECS instances](#).

Context

Custom ID-based machine groups offer significant benefits in the following scenarios:

- If your servers reside in multiple custom network environments such as virtual private clouds (VPCs), some IP addresses of the servers may conflict. In this case, Logtail cannot collect logs as expected. You can use a custom ID to prevent this issue.
- If you want to add multiple servers to a machine group, you can set the same custom ID for new servers as the machine group. Log Service identifies the custom ID and adds the servers with the same custom ID to the machine group.

Procedure

1. Create a file named `user_defined_id` in a specified directory.
 - Linux: Store the file in the `/etc/ilogtail/user_defined_id` directory.
 - Windows: Store the file in the `C:\LogtailData\user_defined_id` directory.
2. Set a custom ID for the server.

Note

- You cannot add Windows and Linux servers to the same machine group. You cannot set the same custom ID for Linux and Windows servers.
- You can set one or more custom IDs for a single server and separate custom IDs with line feeds.
- In the Linux server, if the `/etc/ilogtail/` directory or the `/etc/ilogtail/user_defined_id` file does not exist, you can create the directory and file. In the Windows server, if the `C:\LogtailData` directory or the `C:\LogtailData\user_defined_id` file does not exist, you can also create the directory and file.

- Linux:

Set the custom ID in the `/etc/ilogtail/user_defined_id` file. For example, if you want to set the custom ID to `userdefined`, run the following command to edit the file. In the file, enter `userdefined`.

```
vim /etc/ilogtail/user_defined_id
```

- Windows:

Set the custom ID in the `C:\LogtailData\user_defined_id` file. For example, if you want to set the custom ID to `userdefined_windows`, enter `userdefined_windows` in the `C:\LogtailData\user_defined_id` file.

3. Create a machine group.
 - i. [Log on to the Log Service console](#).
 - ii. In the Projects section, click the name of the project in which you want to create a machine group.
 - iii. In the left-side navigation pane, click the **Machine Groups** icon.
 - iv. Click the  icon next to **Machine Groups**, and then select **Create Machine Group**.

- v. In the Create Machine Group panel, set the parameters. The following table describes the parameters.

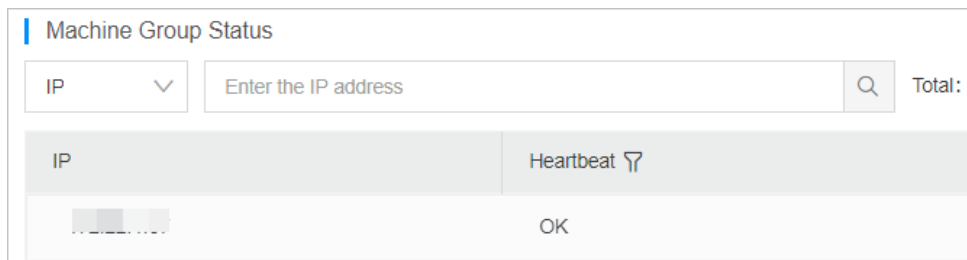
Parameter	Description
Name	The name of the machine group. The name must be 2 to 128 characters in length, and can contain only lowercase letters, digits, hyphens (-), and underscores (_). The name must start and end with a lowercase letter or a digit.  Notice After the machine group is created, you cannot modify its name. Proceed with caution.
Identifier	The identifier of the server. Select Custom ID .
Topic	The topic of the machine group. This topic is used to differentiate log data that is generated in different servers. For more information, see Log topic .
Custom Identifier	Enter the custom ID that is set in .

- vi. Click **OK**.

 **Note** If you need to add a server to the machine group, set a custom ID of the server to the custom ID of the machine group.

4. In the Machine Groups list, click the name of the machine group to view the status of the machine group.

On the **Machine Group Settings** page, you can view the IP address list of the servers in the machine group and the heartbeat status in the **Machine Group Status** section.



The screenshot shows the 'Machine Group Status' section. At the top, there is a search bar with a dropdown menu set to 'IP' and a text input field labeled 'Enter the IP address'. To the right of the input field is a search icon and the text 'Total:'. Below the search bar is a table with two columns: 'IP' and 'Heartbeat'. The 'Heartbeat' column has a status icon (a green checkmark). At the bottom of the table, there is an 'OK' button.

- The Machine Group Status section lists the IP addresses of the servers whose custom ID is the same as the custom ID that you set for the machine group.

For example, the custom ID of a machine group is userdefined and the IP addresses in the Machine Group Status section are 10.10.10.10, 10.10.10.11, and 10.10.10.12. This indicates that you specified the same custom ID for the servers in this 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, set the custom ID to userdefined for the server. Then, you can view the IP address of the server that you added in the Machine Group Status section.

- If the Heartbeat status is OK, the server is connected to Log Service. If the status is FAIL, see [What can I do if no heartbeat packet is received from a Logtail client?](#).

Disable a custom ID

If you want to set the Identifier parameter to IP Addresses, delete the user_defined_id file. The new configurations take effect within 1 minute.

- Linux:

```
rm -f /etc/ilogtail/user_defined_id
```

- In Windows, run the following command:

```
del C:\LogtailData\user_defined_id
```

Time to take effect

By default, after you add, delete, or modify the `user_defined_id` file, the new configurations take effect within 1 minute. If you want the configurations to immediately take effect, run the following command to restart Logtail.

- Linux:

```
/etc/init.d/ilogtaild stop  
/etc/init.d/ilogtaild start
```

- Windows:

- Choose **Start Menu > Control Panel > Administrative Tools > Services**.
- In the **Services** window, select the required service.
 - For Logtail V0.x.x.x, select **LogtailWorker**.
 - For Logtail V1.0.0.0 or later, select **LogtailDaemon**.
- Right-click the service and then select **Restart** to validate the configurations.

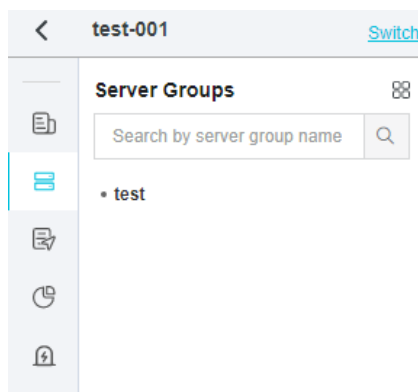
5.3.1.3.5. View server groups

This topic describes how to view the server groups of a project on the **Server Groups** page in the Log Service console.

Procedure

- Log on to the [Log Service console](#).
- Find the target project in the project list and click the project name.
- 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.



5.3.1.3.6. 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

- Log on to the [Log Service console](#).
- Find the target project in the project list and click the project name.
- 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**.

 **Note** The name of the server group cannot be modified.

5. Modify the parameters of the server group, and then click **Save**.

5.3.1.3.7. 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

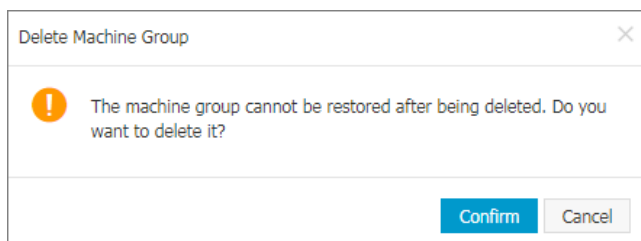
1. [Log on to the Log Service console](#).
2. Find the target project in the project list and click the project name.
3. In the left-side navigation pane of the page that appears, click the **Server Groups** icon to display the list of server groups.
4. Click the name of the server group. On the **Server 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 no heartbeat packet is received from a Logtail client?](#)

5.3.1.3.8. 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**.



5.3.1.3.9. 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.

5.3.1.4. Collect text logs

5.3.1.4.1. Configure text log collection

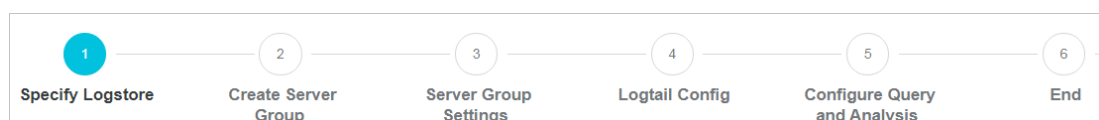
This topic describes the configuration process and collection modes when you use Logtail to collect text logs from servers.

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.



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. 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**.

3. 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.

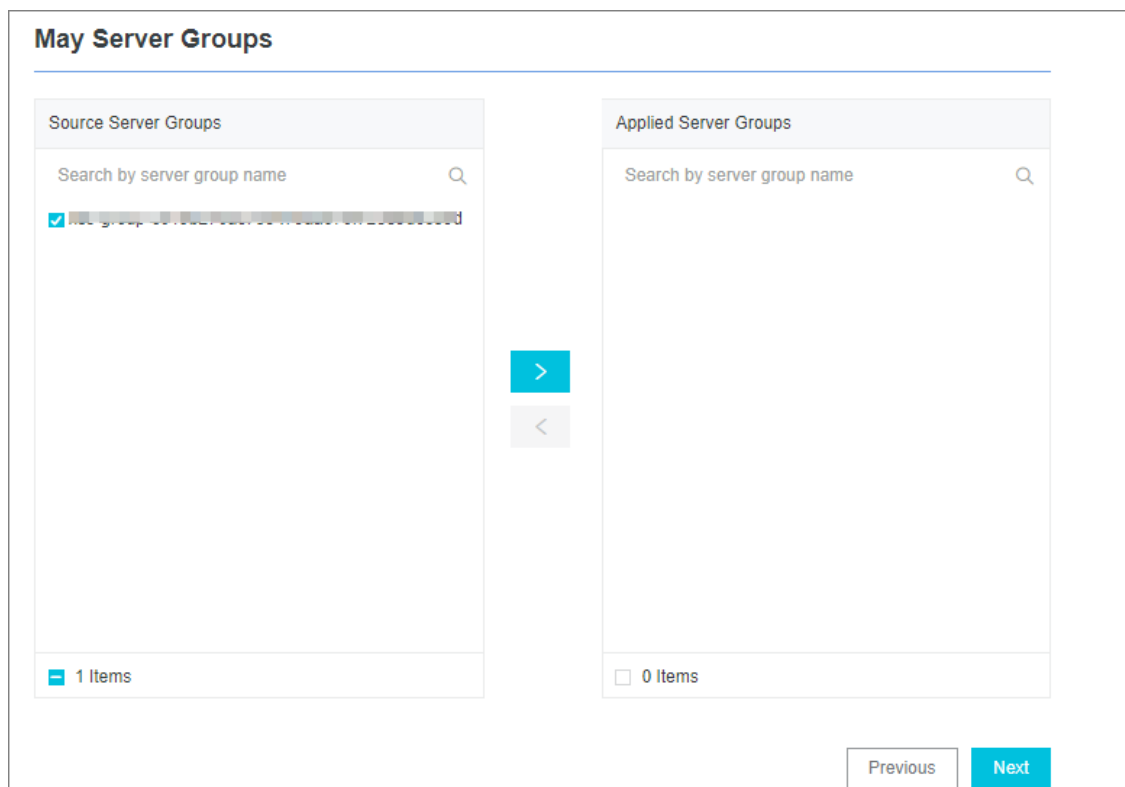
4. Create a machine group and click **Next**.

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. 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 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**.



 **Notice** 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 <code>/apsara/nuwa/.../*.log</code>, Log Service matches the files whose name is suffixed by <code>.log</code> in the <code>/apsara/nuwa</code> directory and its recursive subdirectories. ◦ If you specify <code>/var/logs/app_*/*.log</code>, Log Service matches the files that meet the following conditions: The file name contains <code>.log</code>. The file is stored in a subdirectory of the <code>/var/logs</code> directory or in a recursive subdirectory of the subdirectory. The name of the subdirectory matches the <code>app_*</code> 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 <code>log.log</code> file from the <code>/home/log/nginx/log/log.log</code> directory. You can run the following command to create a symbolic link that points to the 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. <pre>ln -s /home/log/nginx/log /home/log/nginx/link_log</pre> ◦ 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 <code>/tmp/mydir</code> 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 <code>/tmp/mydir/file</code> in the Content column, only the specified file is skipped.

Parameter	Description
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 Collect Kubernetes logs .
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.
Regex to Match First Line	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. 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

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.  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 <code>__raw__</code> 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 the Custom 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 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: ◦ <code>utf8</code>: UTF-8 encoding format ◦ <code>gbk</code>: 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.

Parameter	Description
Filter Configuration	Only logs that meet all filter conditions are collected. Examples: - Collect logs that meet specified conditions: If you set Key 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 .*^(?!.*(healthcheck)).*, 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 [Enable the index feature and configure indexes for a Logstore](#).

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.

5.3.1.4.2. 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.

Context

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 Log Service console.

2. Select a data source.

Select **Single Line - Text Log**.

3. 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.

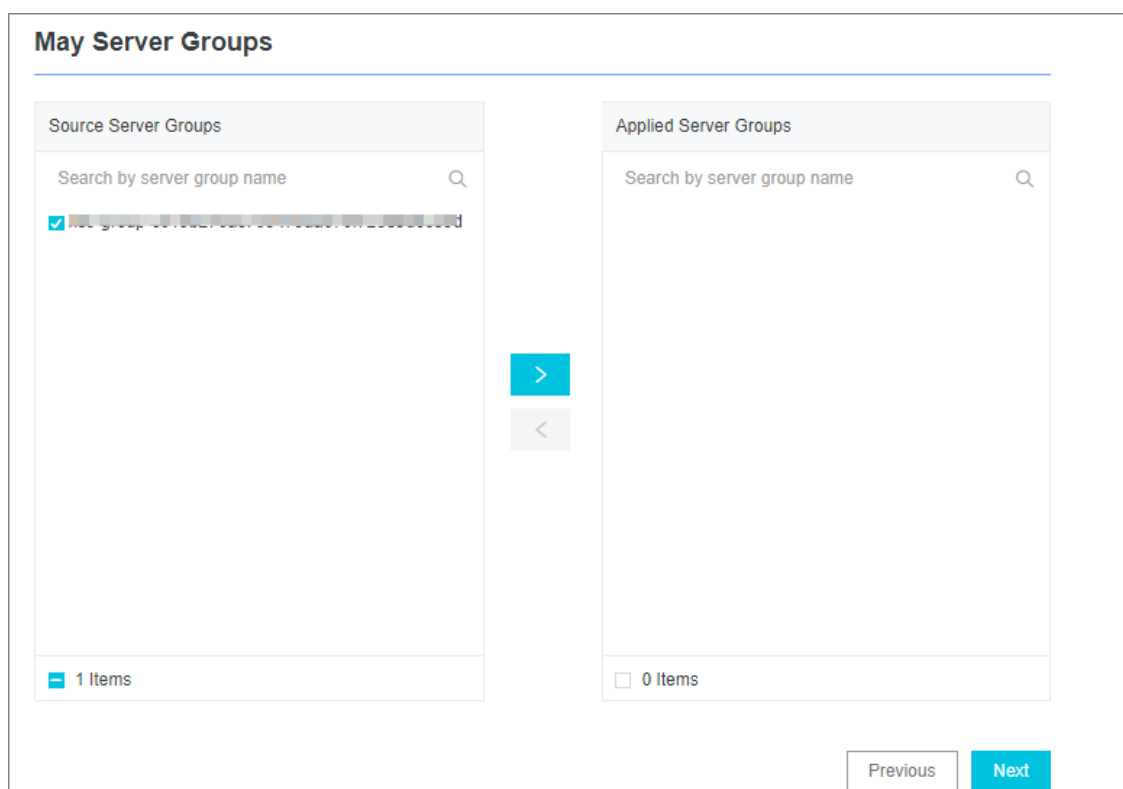
4. Create a machine group and click **Next**.

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. 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 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**.



Notice 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.

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.  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. - 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 <code>/apsara/nuwa/ **/*.log</code>, Log Service matches the files whose name is suffixed by <code>.log</code> in the <code>/apsara/nuwa</code> directory and its recursive subdirectories. - If you specify <code>/var/logs/app_* .../*.log*</code>, Log Service matches the files that meet the following conditions: The file name contains <code>.log</code>. The file is stored in a subdirectory of the <code>/var/logs</code> directory or in a recursive subdirectory of the subdirectory. The name of the subdirectory matches the <code>app_*</code> pattern.  Note - By default, each log file can be collected by using only one Logtail configuration. - You can use only asterisks (<code>*</code>) and question marks (<code>?</code>) 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) 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

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.  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 <code>__raw__</code> 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 the Custom 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 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: ◦ <code>utf8</code>: UTF-8 encoding format ◦ <code>gbk</code>: 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.

Parameter	Description
Filter Configuration	Only logs that meet all filter conditions are collected. Examples: - Collect logs that meet specified conditions: If you set Key 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 .*^(?!.*(healthcheck)).*, 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 [Enable the index feature and configure indexes for a Logstore](#).

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 in full regex mode.

5.3.1.4.3. 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.

Context

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 test a regular expression?](#).

Procedure

1. [Log on to the Log Service console](#).

2. Select a data source.

Select **RegEx - Text Log**.

3. 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 **Next**.

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. 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 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**.

Notice 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.

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.

Parameter	Description
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 <code>/apsara/nuwa/**/*.log</code>, Log Service matches the files whose name is suffixed by <code>.log</code> in the <code>/apsara/nuwa</code> directory and its recursive subdirectories. - If you specify <code>/var/logs/app_*/*.log</code>, Log Service matches the files that meet the following conditions: The file name contains <code>.log</code>. The file is stored in a subdirectory of the <code>/var/logs</code> directory or in a recursive subdirectory of the subdirectory. The name of the subdirectory matches the <code>app_*</code> 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 <code>log.log</code> file from the <code>/home/log/nginx/log</code> directory. You can run the following command to create a symbolic link that points to the 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. <pre>ln -s /home/log/nginx/log /home/log/nginx/link_log</pre> - 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.

Parameter	Description
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. Log Service automatically generates a regular expression. - Specify a regular expression. You can also enter a regular expression. Click Manual 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 test 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 <code>time</code>. After you specify the <code>time</code> field, click Auto Generate in the Time Conversion Format field to parse the time. For more information, see Configure the time format.
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)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.  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.

Parameter	Description
Upload Raw Log	If you turn on Upload Raw Log , each raw log is uploaded to Log Service as a value of the <code>__raw__</code> 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 the Custom 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 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: ◦ <code>utf8</code>: UTF-8 encoding format ◦ <code>gbk</code>: 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 set Key to <code>level</code> and Regex to <code>WARNING ERROR</code>, only WARNING-level and ERROR-level logs are collected. ◦ Filter out logs that do not meet specified conditions. ■ If you set Key to <code>level</code> and Regex to <code>^(?!.*(INFO DEBUG)).*</code>, INFO-level or DEBUG-level logs are not collected. ■ If you set Key to <code>url</code> and Regex to <code>.^(?!.*(healthcheck)).*</code>, logs whose URLs contain healthcheck are not collected. For example, the log is not collected if the value of the Key field is <code>url</code> and the value of the Value field is <code>/inner/healthcheck/jiankong.html</code>.

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 [Enable the index feature and configure indexes for a Logstore](#).

 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 in full regex mode.

5.3.1.4.4. 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 Log Service console](#).
2. In the **Import Data** section, select **Delimiter Mode - Text Log**.
3. 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 **Next**.

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. 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 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**.

May Server Groups

Source Server Groups

Search by server group name

☒ ...

1 Items

Applied Server Groups

Search by server group name

0 Items

>

<

Previous

Next

Notice 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**.

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. 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. 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:
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.

Parameter	Description
Log Sample	Enter a sample log that is retrieved from a log source in an actual scenario. Example: <pre>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</pre>  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.  Note If you set the Delimiter parameter to Hidden Characters, you must enter a character in the following format: <code>0xHexadecimal ASCII code of the non-printable character</code>. For example, if you want to use the non-printable character whose hexadecimal ASCII code is 01, you must enter <code>0x01</code>.
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.  Note If you set the Quote parameter to Hidden Characters, you must enter a character in the following format: <code>0xHexadecimal ASCII code of the non-printable character</code>. For example, if you want to use the non-printable character whose hexadecimal ASCII code is 01, you must enter <code>0x01</code>.
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 log 11 22 33 44 55 is parsed into the following fields: 11, 22, 33, 44, and 55. You can set the keys to A, B, C, D, and E. ◦ If you turn on Incomplete Entry Upload, 55 is uploaded as the value of the D key when Log Service collects the log 11 22 33 55. ◦ If you turn off Incomplete Entry Upload, the log 11 22 33 55 is dropped because the number of fields parsed from the log does not match the number of the specified keys.

Parameter	Description
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, 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.  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 the Custom 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 users or instances.
Custom RegEx	If you set the Topic Generation Mode parameter to File Path RegEx, you must enter a custom regular expression.

Parameter	Description
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 set Key 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 .^(?!.*(healthcheck)).*, 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 <code>/inner/healthcheck/jiankong.html</code>.

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 [Enable the index feature and configure indexes for a Logstore](#).

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 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?Category=YunOsAccountOpLog&AccessKeyId=*****
**&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=*****
***** HTTP/1.1",200,18204,aliyun-sdk-java
05/May/2016:13:31:23,10.10.*.*, "POST /PutData?Category=YunOsAccountOpLog&AccessKeyId=*****
**&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=*****
***** 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 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?Category=YunOsAccountOpLog&AccessKeyId=*****
*****&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2Bmkd6x7
hAgQ7b1c%3D HTTP/1.1&200&&18204&&aliyun-sdk-java
05/May/2016:13:31:23&10.200.**.**&POST /PutData?Category=YunOsAccountOpLog&AccessKeyId=*****
*****&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=*****
***** 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.

5.3.1.4.5. 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.

Context

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 by line](#) or [Use regular expressions to collect logs](#).

Sample JSON logs:

```
{ "url": "POST /PutData?Category=YunOsAccountOpLog&AccessKeyId=U0Ujpek*****&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2Bmkd6x7hAgQ7b1c%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%2Bmkd6x7hAgQ7b1c%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 Log Service console](#).
2. In the **Import Data** section, select **JSON - Text Log**.
3. 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 **Next**.

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. 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 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**.

May Server Groups

Source Server Groups

Search by server group name

☒ ...

1 Items

Applied Server Groups

Search by server group name

0 Items

>
<

Previous
Next

Notice 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**.

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. 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. 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:
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.

Parameter	Description
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, 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.  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 the Custom 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 users or instances.
Custom RegEx	If you set the Topic Generation Mode parameter to File Path RegEx, you must enter a custom regular expression.

Parameter	Description
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 set Key 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 .^(?!.*(healthcheck)).*, 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 <code>/inner/healthcheck/jiankong.html</code>.

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 [Enable the index feature and configure indexes for a Logstore](#).

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 in JSON mode.

5.3.1.4.6. 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 Log Service console](#).
2. In the **Import Data** section, select **Nginx - Text Log**.
3. 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.

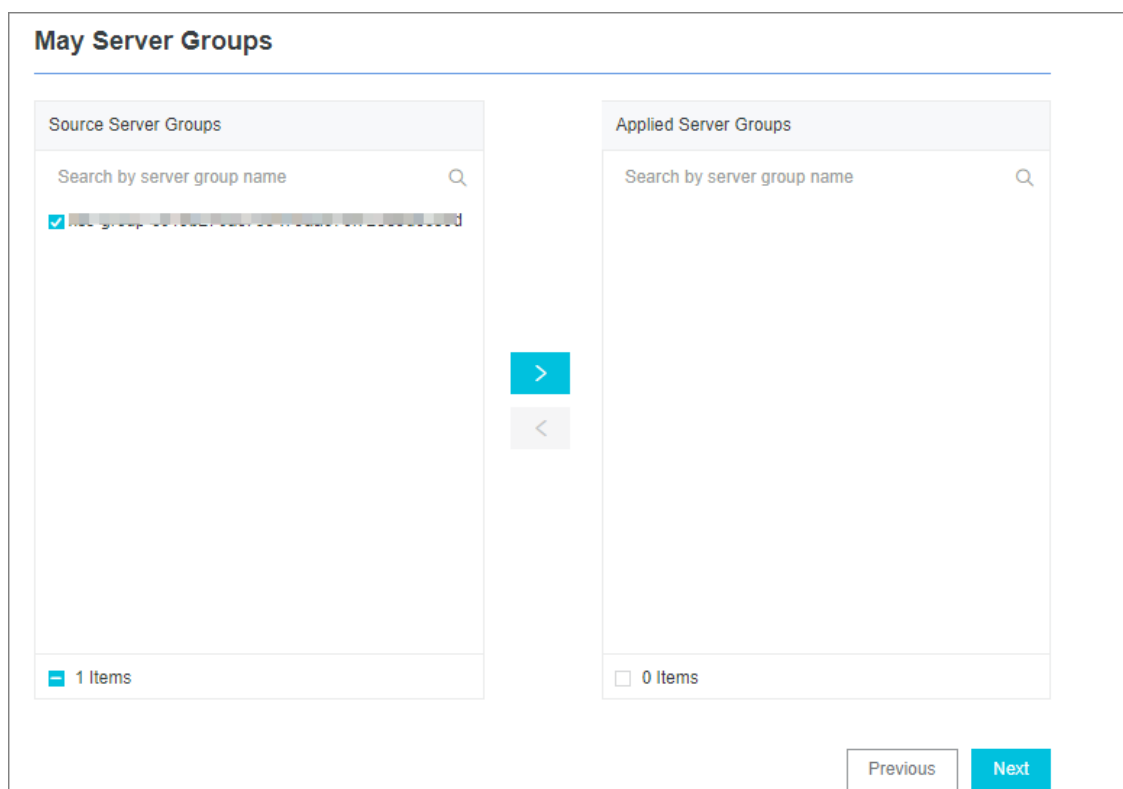
4. Create a machine group and click **Next**.

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. 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 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**.



Notice 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**.

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.  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. 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:
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.
NGINX Log Configuration	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" ' '\$request_time \$request_length ' '\$status \$body_bytes_sent "\$http_referer" ' '"\$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.
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, 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

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.  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 <code>__raw__</code> 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 the Custom 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 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.

Parameter	Description
Filter Configuration	Only logs that meet all filter conditions are collected. Examples: - Collect logs that meet specified conditions: If you set Key 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 .*^(?!.*(healthcheck)).*, 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 <code>/inner/healthcheck/jiankong.html</code>.

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 [Enable the index feature and configure indexes for a Logstore](#).

 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 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.

Log field	Description
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"
```

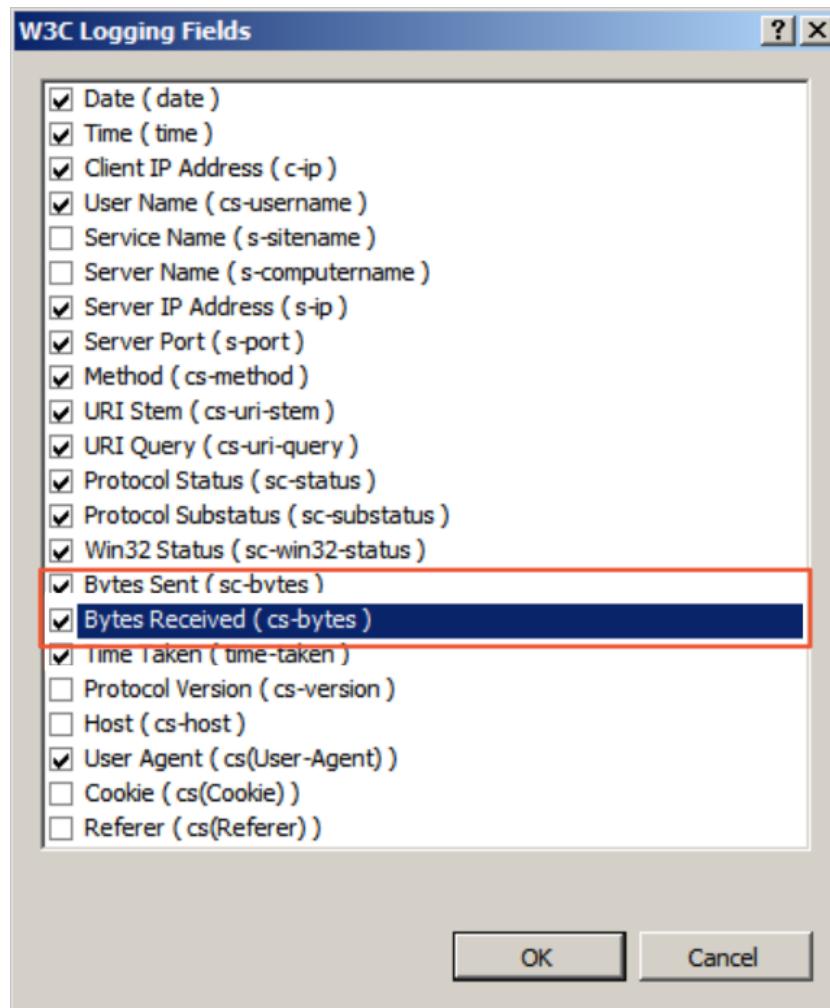
5.3.1.4.7. Collect logs in IIS mode

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 Log Service console.](#)

2. Select a data source.

In the Import Data section, select **IIS - Text Log**.

3. 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 **Next**.

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. 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 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**.

May Server Groups

Source Server Groups

Search by server group name

☒ ...

1 Items

Applied Server Groups

Search by server group name

0 Items

>
<

Previous
Next

Notice If you want to apply a machine group immediately after it is created, the heart beat 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.

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. 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. 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:
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.

Parameter	Description
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 logFile logExtFileFlags field of the IIS configuration file. <pre>logExtFileFlags="Date, Time, ClientIP, UserName, SiteName, ComputerName, ServerIP, Method, UriStem, UriQuery, HttpStatus, Win32Status, BytesSent, BytesRecv, TimeTaken, ServerPort, UserAgent, Cookie, Referer, ProtocolVersion, Host, HttpSubStatus"</pre> ■ Default path of the IIS5 configuration file: C:\WINNT\system32\inetrv\MetaBase.bin ■ Default path of the IIS6 configuration file: C:\WINDOWS\system32\inetrv\MetaBase.xml ■ Default path of the IIS7 configuration file: C:\Windows\System32\inetrv\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, 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

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.  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 <code>__raw__</code> 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 the Custom 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 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: ◦ <code>utf8</code>: UTF-8 encoding format ◦ <code>gbk</code>: 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.

Parameter	Description
Filter Configuration	Only logs that meet all filter conditions are collected. Examples: - Collect logs that meet specified conditions: If you set Key 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 .*^(?!.*(healthcheck)).*, 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 [Enable the index feature and configure indexes for a Logstore](#).

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 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-status 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 <code>domain\username</code>. - 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.

5.3.1.4.8. 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 sample logs](#).

Procedure

1. [Log on to the Log Service console](#).

2. Select a data source.

In the Import Data section, select **Apache - Text Log**.

3. 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.

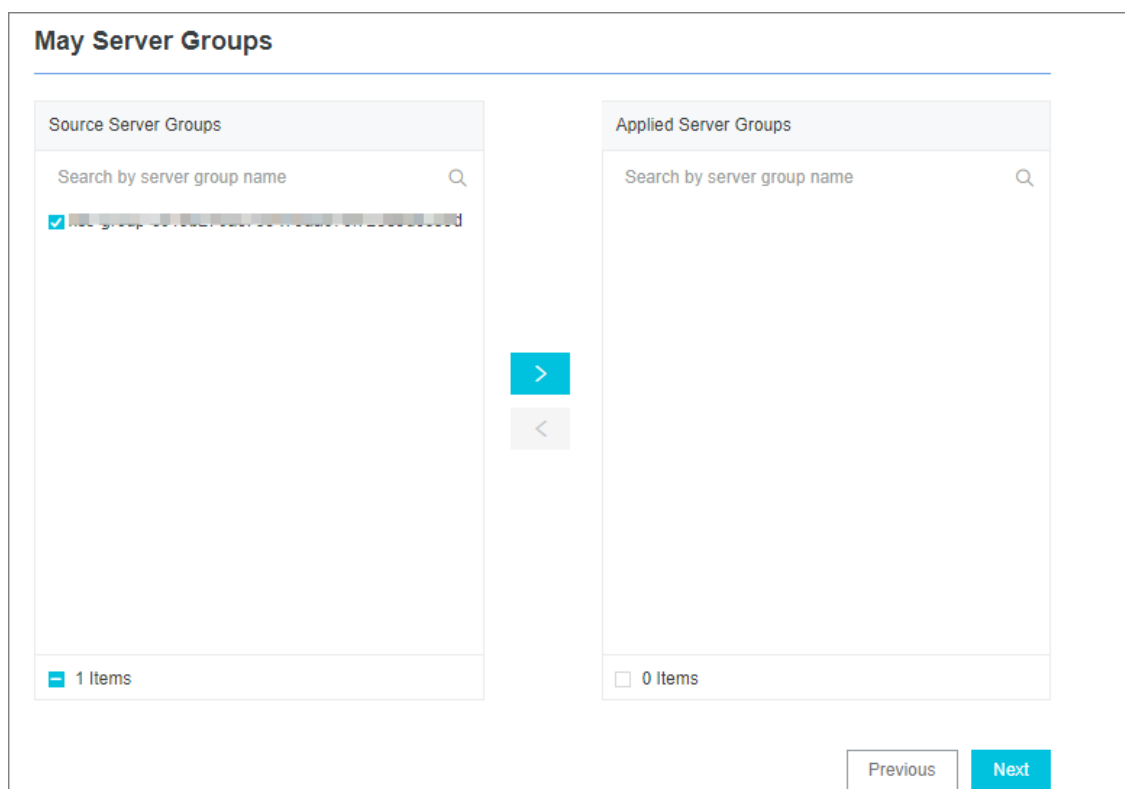
4. Create a machine group and click **Next**.

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. 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 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**.



Notice 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.

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.  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. 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 Service scans all levels of the specified directory to match log files. Examples: ◦ If you specify <code>/apsara/nuwa/**/*.log</code>, Log Service matches the files whose name is suffixed by <code>.log</code> in the <code>/apsara/nuwa</code> directory and its recursive subdirectories. ◦ If you specify <code>/var/logs/app_*/*.log</code>, Log Service matches the files that meet the following conditions: The file name contains <code>.log</code>. The file is stored in a subdirectory of the <code>/var/logs</code> directory or in a recursive subdirectory of the subdirectory. The name of the subdirectory matches the <code>app_*</code> 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 <code>log.log</code> file from the <code>/home/log/nginx/log/log.log</code> directory. You can run the following command to create a symbolic link that points to the 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. <pre>ln -s /home/log/nginx/log /home/log/nginx/link_log</pre> ◦ 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. For more information about container text logs, see Collect container text logs.
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.

Parameter	Description
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 %I %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)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.  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 <code>__raw__</code> field together with the parsed log.

Parameter	Description
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 the Custom 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 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 set Key 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 .*^(?!.*(healthcheck)).*, 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 [Enable the index feature and configure indexes for a Logstore](#).

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 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.
%B	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.
%H	request_protocol_supple	The request protocol.
%I	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.

Format string	Log field	Description
%l	remote_ident	The information that is provided by the remote host for identification.
%m	request_method_supple	The HTTP request method.
%O	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.
%p	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.
"%{Referer}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 Mac OS X 10_11_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.97 Safari/537.36"
```

5.3.1.4.9. 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
0x152436b9a12aecf, 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 `[\d+-\d+-\w+:\d+:\d+,\d+]\s.*`. The following figure shows how to enter a sample log and specify a regular expression in the Log Service console.

Full regex mode

Mode: Full Regex Mode

* 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
0x152436b9a12aecf, 50000
0x152436b9a12aed2, 50000
0x152436b9a12aed1, 50000
0x152436b9a12aed0, 50000
```

* Regex to Match First `[\d+-\d+-\w+:\d+:\d+,\d+]\s[\w+]\s.*`

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.

In this example, the regular expression is `(\S+)\s-\s-\s\[ (\S+)\s[^\]]+\s" (\w+) .*`. The extracted fields are `203.0.113.10`, `13/Mar/2016:10:00`, and `GET`.

- 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.

5.3.1.4.10. 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
%H	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
%p	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 %I:%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
%y	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

Format	Description	Example
%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
2006-01-02T15:04:05.999999999Z07:00	%Y-%m-%dT%H:%M:%S	RFC3339Nano

5.3.1.4.11. 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 in Linux](#) and [Install Logtail in Windows](#).
- 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.

Context

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 [Set Logtail startup parameters](#).

Procedure

- 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$secs-test"
##1.0##log-config-test$metric_system_test"
##1.0##log-config-test$redis-status"
```

- Add a local event.
 - 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:

```
[
  {
    "config" : "${your_config_unique_id}",
    "dir" : "${your_log_dir}",
    "name" : "${your_log_file_name}"
  },
  {
    ...
  }
  ...
]
```

 **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: <i>/data/logs</i> .  Note The directory cannot end with a forward slash (/).
name	The name of the historical log file. The name can contain wildcards. Example: <i>access.log.2018-08-08</i> and <i>access.log*</i> .

The following example shows how to configure a local event in Linux by using the `cat /usr/local/ilogtail/local_event.json` command:

```
[
  {
    "config": "##1.0##log-config-test$ecs-test",
    "dir": "/data/log",
    "name": "access.log*"
  },
  {
    "config": "##1.0##log-config-test$tmp-test",
    "dir": "/tmp",
    "name": "access.log.2017-08-09"
  }
]
```

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.
 - The configurations of the local event in the *local_event.json* file are invalid.
 - The log file does not exist in the path that is specified in the Logtail configuration.
 - The log file has been collected by Logtail.

5.3.1.4.12. 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 `\/(.*)\/serviceA\/.*` regular expression. Then, Log Service generates the following topics for logs that are in different directories: userA, userB, and userC.

```
/logs
| - /userA/serviceA
|   | - service.log
| - /userB/serviceA
|   | - service.log
| - /userC/serviceA
|   | - 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 `?P<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:

```
/home/admin/serviceA/userB/access.log
\\home\\admin\\(?P<service>[^\\/]+) / (?P<user>[^\\/]+) / .*
```

The following custom tags are created for logs:

```
"__tag__ : service : serviceA"
"__tag__ : user : userB"
```

- Static topic generation

You can set **Topic Generation Mode** to **File Path RegEx**. In the **Custom RegEx** field, enter `customized:// + custom topic name`.

 **Note** Static topic generation is supported by Logtail V0.16.21 (Linux) and later.

5.3.1.5. Collect container logs

5.3.1.5.1. 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

Item	CRD	Log Service console
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. [Collect Kubernetes logs.](#)

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. [Collect Kubernetes logs.](#)

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.](#)

5.3.1.5.2. Install the Logtail component

This topic describes how to install the Logtail component in a Kubernetes cluster.

Context

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 `k8s-group-${your_k8s_cluster_id}` and a Logstore named `config-operation-log` are automatically created in the project that you use.

 **Notice** Do not delete the `config-operation-log` Logstore.

Install the Logtail component when you create an ACK cluster

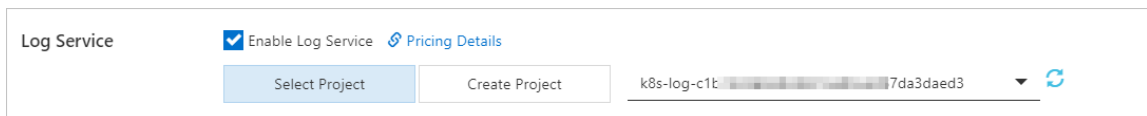
1. [Log on to the Log Service console](#).
2. In the left-side navigation pane, click **Clusters**.
3. On the **Clusters** page, click **Create Kubernetes Cluster**.
4. In the **Component Configurations** step, select **Enable Log Service**.

 **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:

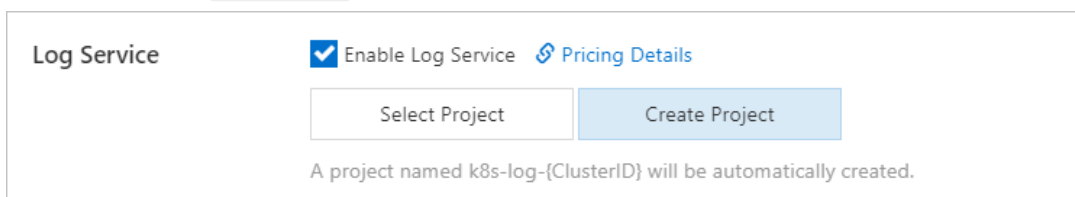
◦ Select Project

You can select an existing project to manage the container logs that are collected.



◦ Create Project

Log Service automatically creates a project named `k8s-log-{ClusterID}` to manage the container logs that are collected. `ClusterID` indicates the unique ID of the ACK cluster that is created.



After the component is installed, a machine group named `k8s-group-${your_k8s_cluster_id}` and a Logstore named `config-operation-log` are automatically created in the project that you use.

 **Notice** Do not delete the `config-operation-log` Logstore.

Self-managed Kubernetes clusters

1. [Log on to the Log Service console](#)
2. Create a project whose name starts with `k8s-log-custom-`.

Example: `k8s-log-custom-sd89ehdq`. For more information, see [Create a project..](#)

3. Log on to your Kubernetes cluster.
4. Run the following command to install the alibaba-log-controller component.

 **Notice** Make sure that the kubectl command-line tool is installed on the machine on which you want to run the command.

- i. Download the `alicloud-log-k8s-custom-install.sh` script.
- ii. Upload the script to the machine.
- iii. Go to the directory of the script and modify permissions.

```
chmod 744 ./alicloud-log-k8s-custom-install.sh;
```

- iv. Run the following installation command:

```
./alicloud-k8s-log-installer.sh --cluster-id ${your_k8s_cluster_id} --ali-uid ${your_ali_uid}
--region-id ${your_k8s_cluster_region_id} {access-key-id} {access-key-secret}
```

You can configure the parameters in the command based on your business requirements. The following table describes the parameters.

Parameter	Description
<code>your_k8s_cluster_id</code>	The ID of your Kubernetes cluster.
<code>your_ali_uid</code>	The ID of your Alibaba Cloud account.  Note The ID of an Alibaba Cloud account is a string. For more information about how to view the ID of an Alibaba Cloud account, see Configure an account ID on a server .
<code>your_k8s_cluster_region_id</code>	The ID of the region where your Kubernetes cluster resides.
<code>access-key-id</code>	The AccessKey ID of your Alibaba Cloud account. For more information, see Obtain an AccessKey pair .
<code>access-key-secret</code>	The AccessKey secret of your Alibaba Cloud account. For more information, see Obtain an AccessKey pair .

After the component is installed, a machine group named `k8s-group-${your_k8s_cluster_id}` and a Logstore named `config-operation-log` are automatically created in the project that you use.

 **Notice**

- 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/overlay2/3fd0043af174cb0273c3c7869500f9e2bdb95d13b1e110172ef57
fe840c82155/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44
beab6e69718/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640
b1e16c22dbe/merged: must be superuser to unmount
.....
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 `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 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:

```
kubect1 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-wm7lw	1/1	Running	0	4d

- Run the following command to view the version number and IP address of Logtail:

```
kubect1 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.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 `kubect1 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] [INFO] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:104] logtai
l plugin Resume:start
[2018-02-05 06:09:02.168807] [INFO] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:106] logtai
l plugin Resume:succcess
[2018-02-05 06:09:02.168822] [INFO] [9] [build/release64/sls/ilogtail/EventDispatcher.cpp:369] star
t 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.

```
kubect1 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's 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](#).

5.3.1.5.3. 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.
 - 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.

4. Click **Use Existing Machine Groups**.

After you install the Logtail component, Log Service automatically creates a machine group named `k8s-group-p-${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**.



Notice 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.

Notice

- 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 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 (<code>^</code>) and ends with a dollar sign (<code>\$</code>) 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>^(nginx 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.

Parameter	Description
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 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 (\$) 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>^(nginx 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 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 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 <code>NGINX_SERVICE_PORT</code> and set the EnvValue parameter to <code>^(80 6379)\$</code>, 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 matched.

Parameter	Description
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 (<code>^</code>) and ends with a dollar sign (<code>\$</code>) for the EnvValue parameter, regular expression matching is performed. For example, if you set the EnvKey parameter to <code>NGINX_SERVICE_PORT</code> and set the EnvValue parameter to <code>^(80 6379)\$</code>, 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.

- 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 <code>^(nginx-log-demo.*)\$</code> , 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 <code>^(default nginx)\$</code> , all containers in the nginx and default namespaces are matched.

Parameter	Description
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 <code>^(container-test)\$</code>, 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 values of the LabelValue parameter. If you specify a value that starts with a caret (<code>^</code>) and ends with a dollar sign (<code>\$</code>), regular expression matching is performed. For example, if you set the LabelKey parameter to <code>app</code> and set the LabelValue parameter to <code>^(test1/test2)\$</code>, containers whose Kubernetes labels consist of <code>app:test1</code> or <code>app:test2</code> 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.
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 (<code>^</code>) and ends with a dollar sign (<code>\$</code>), regular expression matching is performed. For example, if you set the LabelKey parameter to <code>app</code> and set the LabelValue parameter to <code>^(test1/test2)\$</code>, 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 <code>VERSION</code> and set the EnvValue parameter to <code>env_version</code> , Log Service adds the <code>__tag__:__env_version__: v1.0.0</code> field to logs if the environment variable configurations of a container include <code>VERSION=v1.0.0</code> .
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 <code>app</code> and set the LabelValue parameter to <code>k8s_label_app</code> , Log Service adds the <code>__tag__:__k8s_label_app__: serviceA</code> field to logs if the label configurations of a Kubernetes cluster include <code>app=serviceA</code> .

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.

```

"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://:80",
  "HTTP_SVC_PORT=tcp://:80",
  "HTTP_SVC_PORT_80_TCP_PORT=80",
  "NGINX_SERVICE_PORT=80",
  "KUBERNETES_PORT_443_TCP=tcp://:443",
  "KUBERNETES_PORT_443_TCP_PROTO=tcp",
  "HTTP_SVC_SERVICE_PORT=80",
  "KUBERNETES_PORT_443_TCP_ADDR=171.19.134.1",
  "HTTP_SVC_PORT_80_TCP=tcp://:80",

```

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 by line](#).

* Config Name: [Import Other Configuration](#)

* Log Path:

All files under the specified folder (including all directory levels) that conform to the file name convention will be monitored. The file name can be a complete name or a name that contains wildcards. The Linux file path must start with "/"; for example, /apsara/nuwa/.../app.Log. The Windows file path must start with a drive; for example, C:\Program Files\Intel\...*.Log.

Blacklist: ☐

You can configure a blacklist to skip the specified directories or files during log data collection. The names of the specified directories and files support exact match and wildcard match. For example, if you specify the /tmp/mydir directory as a filtering condition, you can skip all files in the directory. If you specify the /tmp/mydir/file directory as a filtering condition, you can skip only the specified file in the directory. [Documentation](#)

Docker File: ☒

For a Docker file, you can directly configure the log path and container tags. Container tags are specified by the configuration of the label whitelist and blacklist and environment variable whitelist and blacklist. Logtail will automatically monitor the creation and destruction of containers, and collect log entries of the specified containers according to the specified tags. For more information, see [Documentation](#)

Label Whitelist:

LabelKey	LabelValue	Delete
Collect the logs from Docker container in the whitelist (empty means collect all logs)		

Label Blacklist:

LabelKey	LabelValue	Delete
Do not collect logs from Docker containers in the blacklist (empty means collecting all logs)		

Environment Variable Whitelist:

EnvKey	EnvValue	Delete
NGINX_PORT_80_TCP_PORT	80	☒
Collects log entries that contain the environment variables in the whitelist. If the whitelist is empty, all log entries will be collected.		

Environment Variable Blacklist:

EnvKey	EnvValue	Delete
POD_NAMESPACE	kube-system	☒

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.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": "ad00a078-4182-██████████",  
    "io.kubernetes.sandbox.id": "511e7d9c3eb6-62f93a29ab0a-f8c8-819dcccda4-dadccccc1dfa6da112969",  
    "maintainer": "NGINX Docker Maintainers <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 by line](#).

* Config Name:

docker-file

[Import Other Configuration](#)

* Log Path:

/var/log/nginx

/**/

access.log

All files under the specified folder (including all directory levels) that conform to the file name convention will be monitored. The file name can be a complete name or a name that contains wildcards. The Linux file path must start with "/"; for example, /apsara/nuwa/.../app.Log. The Windows file path must start with a drive; for example, C:\Program Files\Intel\...*.Log.

Blacklist:

☐

You can configure a blacklist to skip the specified directories or files during log data collection. The names of the specified directories and files support exact match and wildcard match. For example, if you specify the /tmp/mydir directory as a filtering condition, you can skip all files in the directory. If you specify the /tmp/mydir/file directory as a filtering condition, you can skip only the specified file in the directory. [Documentation](#)

Docker File:

☒

For a Docker file, you can directly configure the log path and container tags. Container tags are specified by the configuration of the label whitelist and blacklist and environment variable whitelist and blacklist. Logtail will automatically monitor the creation and destruction of containers, and collect log entries of the specified containers according to the specified tags. For more information, see [Documentation](#)

Label Whitelist:

LabelKey +	LabelValue	Delete
io.kubernetes.container.name	nginx	☒

Collect the logs from Docker container in the whitelist (empty means collect all logs)

Label Blacklist:

LabelKey +	LabelValue	Delete
type	pre	☒

Do not collect logs from Docker containers in the blacklist (empty means collecting all logs)

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 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

- Obtain information about namespaces.

```
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-bxl79
    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
spec:
  containers:
    - args:
        - --log-type=nginx
        - --stdout=true
        - --total-count=1000000000000
        - --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: nginx-log-demo-0
  resources:
```

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 by line](#).

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 by line](#).

Default fields

The following table describes the fields that are included by default in each container text log.

Log field	Description
<code>_image_name_</code>	The name of the image.
<code>_container_name_</code>	The name of the container.
<code>_pod_name_</code>	The name of the pod.
<code>_namespace_</code>	The namespace of the pod.
<code>_pod_uid_</code>	The unique identifier of the pod.
<code>_container_ip_</code>	The IP address of the pod.

5.3.1.5.4. 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 [Collect Kubernetes logs](#).
- 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.
 - 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 in Linux](#).
- 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 [Configure data processing methods](#).

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**.

 **Notice** 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:

```
{
  "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` 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.

 Notice

- 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	Type	Required	Description
<code>IncludeLabel</code>	Map (The values of the <code>LabelKey</code> and <code>LabelValue</code> 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 <code>LabelKey</code> parameter is required, and the <code>LabelValue</code> parameter is optional. ■ If the <code>LabelValue</code> parameter is empty, containers whose container labels contain the keys specified by <code>LabelKey</code> are matched. ■ If the <code>LabelValue</code> parameter is not empty, containers whose container labels consist of the key-value pairs specified by <code>LabelKey</code> and <code>LabelValue</code> are matched. By default, string matching is performed for the values of the <code>LabelValue</code> parameter. Containers are matched only if the values of the container labels are the same as the values of the <code>LabelValue</code> parameter. If you specify a value that starts with a caret (<code>^</code>) and ends with a dollar sign (<code>\$</code>) for the <code>LabelValue</code> parameter, regular expression matching is performed. For example, if you set the <code>LabelKey</code> parameter to <code>io.kubernetes.container.name</code> and set the <code>LabelValue</code> parameter to <code>^(nginx 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.

Parameter	Type	Required	Description
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 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 (<code>^</code>) and ends with a dollar sign (<code>\$</code>) 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>^(nginx/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 filtered out.

Parameter	Type	Required	Description
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 (<code>^</code>) and ends with a dollar sign (<code>\$</code>) for the EnvValue parameter, regular expression matching is performed. For example, if you set the EnvKey parameter to <code>NGINX_SERVICE_PORT</code> and set the EnvValue parameter to <code>^(80 6379)\$</code>, 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 matched.

Parameter	Type	Required	Description
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 caret (<code>^</code>) and ends with a dollar sign (<code>\$</code>) for the EnvValue parameter, regular expression matching is performed. For example, if you set the EnvKey parameter to <code>NGINX_SERVICE_PORT</code> and set the EnvValue parameter to <code>^(80 6379)\$</code>, 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.

- 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	Type	Required	Description

Parameter	Type	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 Kubernetes labels are the same as the values of the LabelValue parameter. If you specify a value that starts with a caret (<code>^</code>) and ends with a dollar sign (<code>\$</code>), regular expression matching is performed. For example, if you set the LabelKey parameter to <code>app</code> and set the LabelValue parameter to <code>^(test1/test2)\$</code>, containers whose Kubernetes labels consist of <code>app:test1</code> or <code>app:test2</code> 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.

Parameter	Type	Required	Description
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 Kubernetes labels are the same as the values of the LabelValue parameter. If you specify a value that starts with a caret (<code>^</code>) and ends with a dollar sign (<code>\$</code>), regular expression matching is performed. For example, if you set the LabelKey parameter to <code>app</code> and set the LabelValue parameter to <code>^(test1/test2)\$</code>, containers whose Kubernetes labels consist of <code>app:test1</code> or <code>app:test2</code> 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 filtered out.
K8sNamespaceRegex	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": "<code>^(default nginx)\$</code>", 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": "<code>^(nginx-log-demo.*)\$</code>", all containers in the pod whose name starts with <code>nginx-log-demo</code> are matched.

Parameter	Type	Required	Description
K8sContainerRegex	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	Type	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 <code>VERSION</code> and set the EnvValue parameter to <code>env_version</code> , Log Service adds the <code>__tag__:__env_version__:v1.0.0</code> field to logs if the environment variable configurations of a container include <code>VERSION=v1.0.0</code> .
ExternalK8sLabelTag	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 <code>app</code> and set the LabelValue parameter to <code>k8s_label_app</code> , Log Service adds the <code>__tag__:__k8s_label_app__:serviceA</code> field to logs if the label configurations of a Kubernetes cluster include <code>app=serviceA</code> .

- Other parameters

Parameter	Type	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.

Parameter	Type	Required	Description
BeginLineTimeoutMs	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.
BeginLineCheckLength	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://:80",
  "HTTP_SVC_PORT=tcp://:80",
  "HTTP_SVC_PORT_80_TCP_PORT=80",
  "NGINX_SERVICE_PORT=80",
  "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",

```

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

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.

```

"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/ad00a078-85/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": "ad00a07",
  "io.kubernetes.sandbox.id": "5216-a8d0b6891dfa6da112969",
  "maintainer": "NGINX Docker Maintainers <docker-maint@nginx.com>"
},
"StopSignal": "SIGTERM"

```

2. Create a Logtail configuration.

Example:

```

{
  "inputs": [
    {
      "type": "service_docker_stdout",
      "detail": {
        "Stdout": true,
        "Stderr": true,
        "IncludeLabel": {
          "io.kubernetes.container.name": "nginx"
        }
      }
    }
  ]
}

```

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.

```

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-bxl79
    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
spec:
  containers:
    - args:
      - --log-type=nginx
      - --stdout=true
      - --total-count=1000000000000
      - --log-file-size=10000000000
      - --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: nginx-log-demo-0

```

2. Create a Logtail configuration.

Example:

```

{
  "inputs": [
    {
      "type": "service_docker_stdout",
      "detail": {
        "Stdout": true,
        "Stderr": true,
        "K8sNamespaceRegex": "^ (default) $",
        "K8sPodRegex": "^ (nginx-log-demo.*) $",
        "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-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.

Example:

```

{
  "inputs": [
    {
      "type": "service_docker_stdout",
      "detail": {
        "Stdout": true,
        "Stderr": true,
        "IncludeK8sLabel": {
          "job-name": "^(nginx-log-demo.*)$"
        }
      }
    }
  ]
}

```

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.DemoCon
troller : service start
2021-02-03 14:18:41.969 ERROR [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoCon
troller : 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.DemoCon
troller : 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, level, 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.

 **Note** You 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 [Configure data processing methods](#).

```
{
  "inputs": [
    {
      "detail": {
        "BeginLineCheckLength": 10,
        "BeginLineRegex": "\\d+-\\d+-\\d+.*",
        "IncludeLabel": {
          "app": "monitor"
        }
      },
      "type": "service_docker_stdout"
    }
  ],
  "processors": [
    {
      "type": "processor_regex",
      "detail": {
        "SourceKey": "content",
        "Regex": "(\\d+-\\d+-\\d+ \\d+:\\d+:\\d+\\.\\d+)\\s+(\\w+)\\s+\\[([\\^]]+)\\]\\s+\\[([\\^]]+)\\s+([\\s\\S]*)",
        "Keys": [
          "time",
          "level",
          "module",
          "thread",
          "message"
        ],
        "NoKeyError": true,
        "NoMatchError": true,
        "KeepSource": false
      }
    }
  ]
}
```

- **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_:monitor
_image_name_:example.com-hangzhou.aliyuncs.aaaaaaaaaaaaaa
_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
<code>_time_</code>	The time at which the data is uploaded. Example: <code>2021-02-02T02:18:41.979147844Z</code> .
<code>_source_</code>	The type of the data source. Valid values: <code>stdout</code> and <code>stderr</code> .
<code>_image_name_</code>	The name of the image.
<code>_container_name_</code>	The name of the container.
<code>_pod_name_</code>	The name of the pod.
<code>_namespace_</code>	The namespace of the pod.
<code>_pod_uid_</code>	The unique identifier of the pod.
<code>_container_id_</code>	The IP address of the pod.

5.3.1.5.5. 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 [Collect Kubernetes logs](#).

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.
3. The alibaba-log-controller sends requests to Log Service to create a Logstore, create a Logtail 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.

4. 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 `/home/logs/app_log/` directory of the host, you must specify `/logtail_host/home/logs/app_log/` 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.

 Notice

- 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                          # The default value is used. You do not need to modify this parameter.
metadata:
  name: simple-stdout-example                  # 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                         # The name of the Logstore. If the Logstore that you specify does not exist, Log Service automatically creates a Logstore.
  shardCount: 2                               # Optional. The number of shards. Valid values: 1 to 10. Default value: 2.
  lifecycle: 90                               # 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          # 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 configuration examples.
    ...

```

Parameter	Type	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.
logstore	string	Yes	The name of the Logstore. 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.

Parameter	Type	Required	Description
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.  Notice 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.
machineGroups	array	No	The machine group to which the Logtail configuration is applied. The default value is the machine group named <code>k8s-group-\${your_k8s_cluster_id}</code>. This machine group is automatically created by Log Service when you install the Logtail component.
logtailConfig	object	Yes	The detailed settings of the Logtail configuration. In most cases, you need to configure only the inputType, configName, and inputDetail parameters.

- 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](#).

 **Notice** 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 aliyunlogconfigs 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/v1alpha1
kind: AliyunLogConfig
metadata:
  annotations:
    kubernetes.io/last-applied-configuration: |
      {"apiVersion":"log.alibabacloud.com/v1alpha1","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"}]},"inputType":"plugin"}}}}
  creationTimestamp: "2021-10-29T08:40:33Z"
  generation: 2
  name: docker-stdout
  namespace: default
  resourceVersion: "1350968"
  uid: 35f9516b-4c71-410a-b8a6-114769fabdc
spec:
  extensions: ""
  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`. To view the environment variables of a container, you can log on to the host on which the container resides. CRD 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:
              # 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.

Collect the access logs of Grafana from containers in simple mode and parse the access logs into structured data by using regular expressions. The environment variable configurations of the container where Grafana resides include `GF_INSTALL_PLUGINS=grafana-piechart-....`. To view the environment variables of the 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 Logtail 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 variable 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.
              Regex: 't=(\d+-\d+-\d+:\d+:\d+) lvl=(\w+) msg="([^\"]+)" logger=(\w+) userId=(\w+) orgId=(\w+) uname=(\S*) method=(\w+) path=(\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: true
              # The settings that allow Logtail to report an error when the specified source field does not exist.
              NoKeyError: true
              # The settings that allow Logtail to report an error when the specified regular expression does not match the value of the specified 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__: [REDACTED]
      __tag__: __hostname__: iZbp1-[REDACTED]-p9rZ
      __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. Example: 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(AbstractFallbackba
ckSQLExceptionTranslator.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 are 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 successfully 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 [REDACTED]
                  _tag_ : hostname : iZbp14 [REDACTED] jp9rZ
                  _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. Example: 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 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: ''

```

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?Category=YunOsAccountOpLog&AccessKeyId=U0Ujpek*****&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2BmKd6x7hAgQ7b1c%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"
}
```

- 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: ""
```

5.3.1.5.6. 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 [Collect Kubernetes logs](#).

Context

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.

 **Notice** 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=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.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
          # when receive 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:
```

```

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
<i><code>\${your_aliyun_user_id}</code></i>	The ID of your Apsara Stack tenant account. For more information, see Configure a user identifier .
<i><code>\${your_machine_group_user_defined_id}</code></i>	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 ID-based machine group .
<i><code>\${your_region_config}</code></i>	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.

 **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
name	The name of the volume. You can specify a name based on your business requirements.  Notice The value of the name parameter in the volumeMounts node and the value of the name 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 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.



Notice

- 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      # The default value is used. You do not need to modify this parameter.
kind: AliyunLogConfig                          # The default value is used. You do not need to modify this parameter.
metadata:
  name: simple-stdout-example                  # 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                        # The name of the Logstore. If the Logstore that you specify does not exist, Log Service automatically creates a Logstore.
  machineGroups:                             # 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. This machine group is used to associate Sidecar with the CRD.
    - nginx-log-sidecar
    shardCount: 2                             # Optional. The number of shards. Valid values: 1 to 10. Default value: 2.
    lifeCycle: 90                             # Optional. 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.
    logtailConfig:                           # The Logtail configuration.
      inputType: file                         # 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.
      configName: simple-stdout-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 detailed settings of the Logtail configuration. For more information, see the following configuration examples.
      ...

```

Parameter	Type	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.
logstore	string	Yes	The name of the Logstore. 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.

Parameter	Type	Required	Description
machineGroups	array	Yes	The name of the machine group. The name must be the same as the value of the <code>your_machine_group_user_defined_id</code> parameter that you configured when you installed Sidecar. Example: <code>nginx-log-sidecar</code>. Log Service creates a machine group to associate Sidecar with the CRD based on the name that you specify.  Notice You must specify a custom identifier for the machine group in the following format: <pre>machineGroups: - nginx-log-sidecar</pre>
logtailConfig	object	Yes	The detailed settings of the Logtail configuration. In most cases, you need to configure only the <code>inputType</code>, <code>configName</code>, and <code>inputDetail</code> parameters.

- 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.
- 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.aliyun.com/repository/cn-hangzhou/log-service/logtail/detai
1
      # this images is released for every region
      image: registry.cn-hangzhou.aliyuncs.com/log-service/logtail:latest
      # 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

 **Notice** 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-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 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.
      regex: '(\S+)\s(\S+)\s\S+\s\S+\s"(\S+)\s(\S+)\s+([^\"]+)"\s+(\S+)\s(\S+)\s(\d+)\s(\d+)\s(\S+)\s"([^\"]+)"\s.*'
      # 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

 **Notice** 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-demo
  # 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/nginx
          ##### logtail sidecar container
        - name: logtail
          # more info: https://cr.console.aliyun.com/repository/cn-hangzhou/log-service/logtail/detai
1
          # this images is released for every region
          image: registry.cn-hangzhou.aliyuncs.com/log-service/logtail:latest
          # 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_|_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
- name: nginx-logs
  mountPath: /var/log/nginxs
##### share this volume
volumes:
- name: nginx-log
  emptyDir: {}
- name: nginx-logs
  emptyDir: {}
```

- CRD configuration example

Create two Logtail configurations to collect NGINX access logs from different directories.

- Collect NGINX access logs from the `/var/log/nginx/access.log` directory

 **Notice** 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: 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-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 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.
      regex: '(\S+)\s(\S+)\s\S+\s\S+(\S+)\s(\S+)\s+([^\s]+)"\s+(\S+)\s(\S+)\s(\d+)\s(\d+)\s(\S+)\s"([^\s]+)"\s.*'
      # The keys that you want to extract from logs.
      key: ["time", "ip", "method", "url", "protocol", "latency", "payload", "status", "response-size", "user-agent"]
```

- Collect NGINX access logs from the `/var/log/nginx/access.log` directory

 **Notice** 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: 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-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 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.
      regex: '(\S+)\s(\S+)\s\S+\s\S+(\S+)\s(\S+)\s+([^\s]+)"\s+(\S+)\s(\S+)\s(\d+)\s(\d+)\s(\S+)\s\S+([^\s]+)"\s.*'
      # 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
```

5.3.1.5.7. 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 [Collect Kubernetes logs](#).

Context

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.

 **Notice** 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=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.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
            # when receive 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
<i><code>\${your_aliyun_user_id}</code></i>	The ID of your Apsara Stack tenant account. For more information, see Configure a user identifier .
<i><code>\${your_machine_group_user_defined_id}</code></i>	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 ID-based machine group .
<i><code>\${your_region_config}</code></i>	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.

 **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
name	The name of the volume. You can specify a name based on your business requirements.  Notice The value of the name parameter in the volumeMounts node and the value of the name 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  **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.  Notice 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 .
Topic	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>your_machine_group_user_defined_id</i> 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](#).

 **Notice** 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](#).

 **Notice** 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.

5.3.1.5.8. 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 `${your_region_name}`, `${your_aliyun_user_id}`, and `${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
<code>\${your_region_name}</code>	The ID of the region where your project resides and the type of the network that your project uses.
<code>\${your_aliyun_user_id}</code>	The ID of your Apsara Stack tenant account. For more information, see Configure a user identifier .
<code>\${your_machine_group_user_defined_id}</code>	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 ID-based machine group .

Note

You can customize the startup parameters of the Logtail container only if the following conditions are met:

- The following environment variables are configured: `ALIYUN_LOGTAIL_USER_DEFINED_ID`, `ALIYUN_LOGTAIL_USER_ID`, and `ALIYUN_LOGTAIL_CONFIG`.
- The `/var/run` directory of the host is mounted on the `/var/run` directory of the Logtail container.
- The root directory of the host is mounted on the `/logtail_host` directory of the Logtail container.
- 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-ABC0-4A92-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
<code>_time_</code>	The point in time when data is uploaded. Example: <code>2018-02-02T02:18:41.979147844Z</code> .

Log field	Description
<code>_source_</code>	The type of the input source. Valid values: <code>stdout</code> and <code>stderr</code> .
<code>_image_name_</code>	The name of the image.
<code>_container_name_</code>	The name of the container.
<code>_container_ip_</code>	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
<code>_image_name_</code>	The name of the image.
<code>_container_name_</code>	The name of the container.
<code>_container_ip_</code>	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 start up 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 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`. 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:10
4]    logtail plugin Resume:start
[2018-02-06 08:13:35.722135]    [INFO]    [8]    [build/release64/sls/ilogtail/LogtailPlugin.cpp:10
6]    logtail plugin Resume:succes
[2018-02-06 08:13:35.722149]    [INFO]    [8]    [build/release64/sls/ilogtail/EventDispatcher.cpp:
369]    start add existed check point events, size:0
[2018-02-06 08:13:35.722155]    [INFO]    [8]    [build/release64/sls/ilogtail/EventDispatcher.cpp:
511]    add existed check point events, size: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:37
76]    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$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fbc2bdb95d13b1e110172ef57
fe840c82155/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44
beab6e69718/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640
b1e16c22dbe/merged: must be superuser to unmount
.....
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
ilogtail is running
```

- **Restart Logtail.**

To restart Logtail, use the following sample code:

```
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
```

5.3.1.5.9. 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&regionId={region-id}
&internal=false&accessKeyId={accessKeyId}&accessKeySecret={accessKeySecret}
        - --sink=sls:https://cn-beijing.log.aliyuncs.com?project=k8s-xxxx&logStore=k8s-event&regionId=cn-beijing&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	Type	Required	Description
endpoint	string	Yes	The Log Service endpoint. For more information, see Manage 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 Manage 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"
  },
  "involvedObject":{
    "kind":"Pod",
    "namespace":"kube-system",
    "name":"logtail-ds-blkkr",
    "uid":"2a360760-****",
    "apiVersion":"v1",
    "resourceVersion":"6129161",
    "fieldPath":"spec.containers{logtail}"
  },
  "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	Type	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.

5.3.1.5.10. 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, Docker 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 [Collect Kubernetes logs](#).

- 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 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 **Next**.

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. 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 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**.

May Server Groups

Source Server Groups

Search by server group name

☒

log group 1

1 Items

Applied Server Groups

Search by server group name

0 Items

>

<

Previous

Next

 **Notice** 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.

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.

Parameter	Description
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.  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.  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.  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.

Parameter	Description
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. 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. - 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.pod.namespace : backend-prod` or `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 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 [Enable the index feature and configure indexes for a Logstore](#).

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.

Configuration examples

- Configure environment variables

Collect the logs of the containers whose environment variables include `NGINX_PORT_80_TCP_PORT=80` and exclude `POD_NAMESPACE=kube-system`. The log file path is `/var/log/nginx/access.log` and logs are parsed in simple mode.

```

"openStdin": false,
"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:// :80",
  "HTTP_SVC_PORT=tcp:// :80",
  "HTTP_SVC_PORT_80_TCP_PORT=80",
  "NGINX_SERVICE_PORT=80",
  "KUBERNETES_PORT_443_TCP=tcp:// :443",
  "KUBERNETES_PORT_443_TCP_PROTO=tcp",
  "HTTP_SVC_SERVICE_PORT=80",
  "KUBERNETES_PORT_443_TCP_ADDR=171.19.137.1",
  "HTTP_SVC_PORT_80_TCP=tcp:// :80",

```

- 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.

[illegible]

Default fields

The following table describes the fields that are uploaded by default for each log of a common Docker container.

Log field	Description
<code>_image_name_</code>	The name of the image.
<code>_container_name_</code>	The name of the container.
<code>_container_ip_</code>	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
<code>_image_name_</code>	The name of the image.
<code>_container_name_</code>	The name of the container.
<code>_pod_name_</code>	The name of the pod.
<code>_namespace_</code>	The namespace where the pod resides.
<code>_pod_uid_</code>	The unique identifier of the pod.
<code>_container_ip_</code>	The IP address of the pod.

5.3.1.5.11. 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.

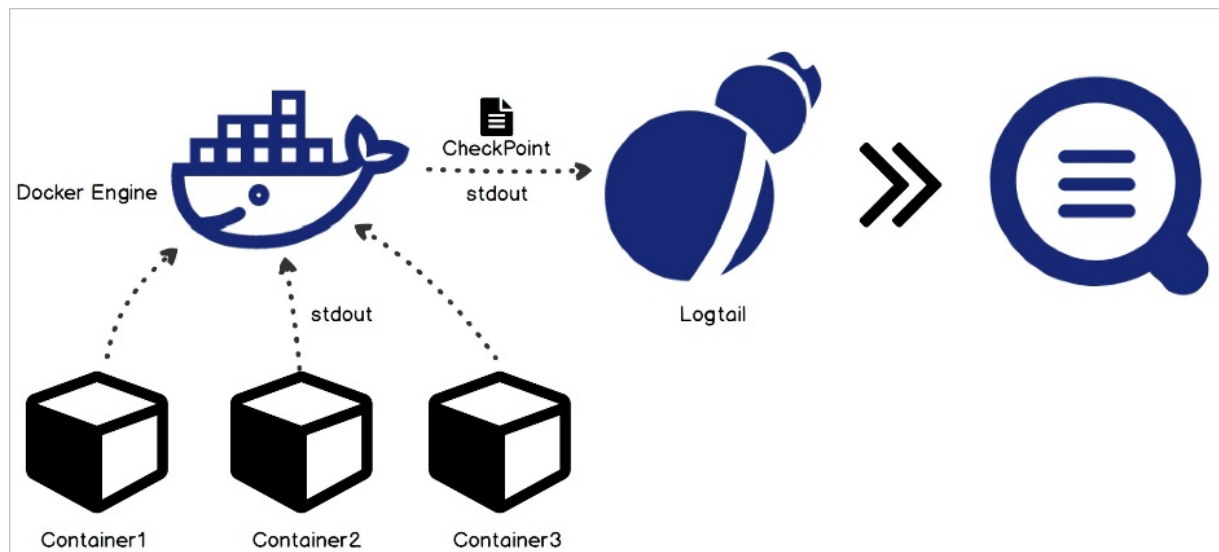
Note

- 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 in Linux](#).
- **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 [Configure data processing methods](#).

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 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 **Next**.

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. 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 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**.

The screenshot shows a web interface titled "May Server Groups". It is divided into two main panels: "Source Server Groups" on the left and "Applied Server Groups" on the right. Both panels have a search bar labeled "Search by server group name". In the "Source Server Groups" panel, there is one item listed with a checkbox that is checked. In the "Applied Server Groups" panel, there are no items listed. Between the two panels, there are two buttons: a blue button with a right-pointing arrow (>) and a grey button with a left-pointing arrow (<). At the bottom of each panel, there is a status bar: "1 Items" for the source and "0 Items" for the applied groups. At the bottom right of the interface, there are two buttons: "Previous" and "Next".

Notice 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. 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"
        }
      }
    }
  ]
}
```

The type of the input source is `service_docker_stdout`.

Parameter	Type	Required	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 istio-system)\$, logs are collected from a container named kube-system and a container named istio-system.

Parameter	Type	Required	Description
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 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 not collected from a container named kube-system or a container named istio-system.

Parameter	Type	Required	Description
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.

Parameter	Type	Required	Description
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-systemistio-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.
BeginLineTimeoutMs	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.
BeginLineCheckLength	int	No	The size of the start part in the first line of a log that matches the specified regular expression. Default value: 10 × 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 × 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.pod.namespace : backend-prod`, the logs of all containers in the pod 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 `io.kubernetes.pod.namespace` and `io.kubernetes.container.name` labels. You can also specify the IncludeEnv or ExcludeEnv parameter based on your business requirements.

7. 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 [Enable the index feature and configure indexes for a Logstore](#).

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.

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
<code>_time_</code>	The point in time when data is uploaded. Example: <code>2018-02-02T02:18:41.979147844Z</code> .
<code>_source_</code>	The type of the input source. Valid values: <code>stdout</code> and <code>stderr</code> .
<code>_image_name_</code>	The name of the image.
<code>_container_name_</code>	The name of the container.
<code>_container_ip_</code>	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
<code>_time_</code>	The point in time when data is uploaded. Example: <code>2018-02-02T02:18:41.979147844Z</code> .
<code>_source_</code>	The type of the input source. Valid values: <code>stdout</code> and <code>stderr</code> .
<code>_image_name_</code>	The name of the image.
<code>_container_name_</code>	The name of the container.
<code>_pod_name_</code>	The name of the pod.
<code>_namespace_</code>	The namespace where the pod resides.
<code>_pod_uid_</code>	The unique identifier of the pod.
<code>_container_id_</code>	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_P` and exclude `POD_NAMESPACE=kube-system`.

Configuration example of environment variables

```

"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://:80",
  "HTTP_SVC_PORT=tcp://:80",
  "HTTP_SVC_PORT_80_TCP_PORT=80",
  "NGINX_SERVICE_PORT=80",
  "KUBERNETES_PORT_443_TCP=tcp://:443",
  "KUBERNETES_PORT_443_TCP_PROTO=tcp",
  "HTTP_SVC_SERVICE_PORT=80",
  "KUBERNETES_PORT_443_TCP_ADDR=17",
  "HTTP_SVC_PORT_80_TCP=tcp://:80",

```

The following script shows the configurations of the environment variables:

```
{
  "inputs": [
    {
      "type": "service_docker_stdout",
      "detail": {
        "Stdout": true,
        "Stderr": true,
        "IncludeEnv": {
          "NGINX_PORT_80_TCP_PORT": "80"
        },
        "ExcludeEnv": {
          "POD_NAMESPACE": "kube-system"
        }
      }
    }
  ]
}
```

- Configure labels

Collect the stdout and stderr logs of the containers whose labels include `io.kubernetes.container.name=nginx` and exclude `type=pre`.

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/ad00a078-85/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": "ad00a078-85",
    "io.kubernetes.sandbox.id": "5216-a8d0b6891dfa6da112969",
    "maintainer": "NGINX Docker Maintainers <docker-maint@nginx.com>"
  },
  "StopSignal": "SIGTERM"
}
```

The following script shows the label configurations:

```
{
  "inputs": [
    {
      "type": "service_docker_stdout",
      "detail": {
        "Stdout": true,
        "Stderr": true,
        "IncludeLabel": {
          "io.kubernetes.container.name": "nginx"
        },
        "ExcludeLabel": {
          "type": "pre"
        }
      }
    }
  ]
}
```

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.DemoCon
troller : service start
2018-02-03 14:18:41.969 ERROR [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoCon
troller : 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.DemoCon
troller : 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.

```
{
  "inputs": [
    {
      "detail": {
        "BeginLineCheckLength": 10,
        "BeginLineRegex": "\\d+-\\d+-\\d+.*",
        "IncludeLabel": {
          "app": "monitor"
        }
      },
      "type": "service_docker_stdout"
    }
  ]
}
```

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:

```
{
  "inputs": [
    {
      "detail": {
        "BeginLineCheckLength": 10,
        "BeginLineRegex": "\\d+-\\d+-\\d+.*",
        "IncludeLabel": {
          "app": "monitor"
        }
      },
      "type": "service_docker_stdout"
    }
  ],
  "processors": [
    {
      "type": "processor_regex",
      "detail": {
        "SourceKey": "content",
        "Regex": "(\\d+-\\d+-\\d+ \\d+:\\d+:\\d+\\.\\d+)\\s+(\\w+)\\s+\\[([^\"]+)]\\s+\\[([^\"]+)]\\s+:\\s+([\\s\\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.
__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
```

- Collect the JSON logs of the containers whose labels include `app=monitor`. The following script shows the configurations of log collection and data processing:

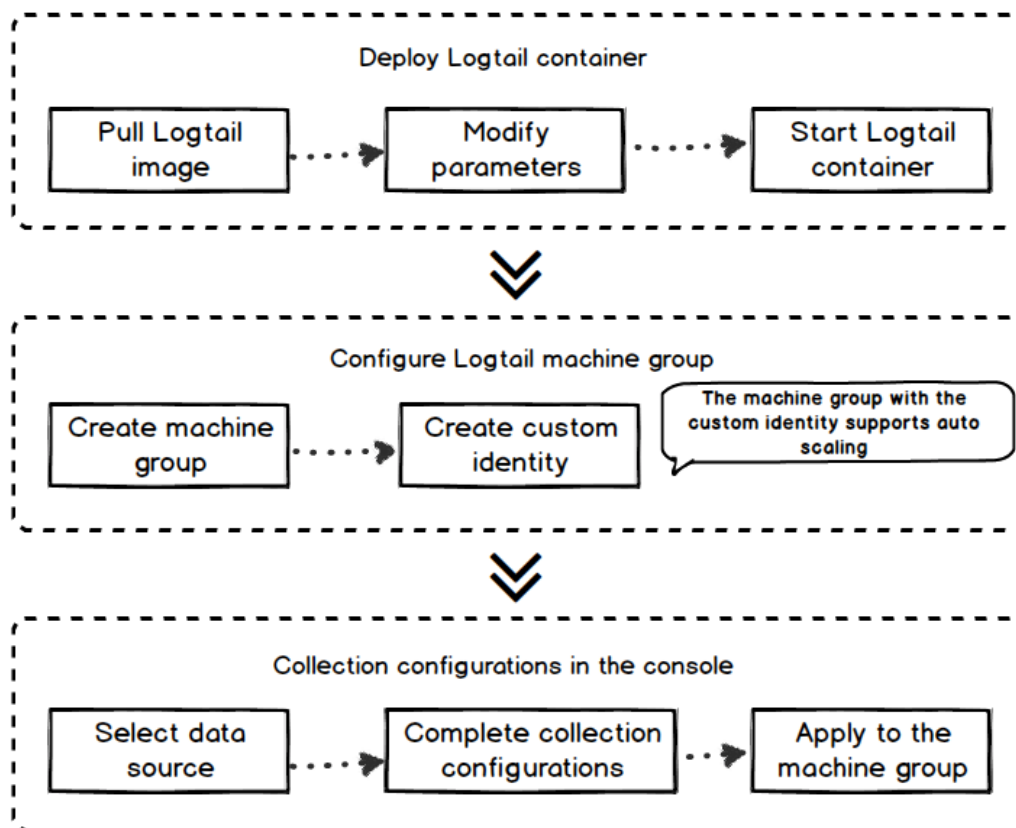
```
{
  "inputs": [
    {
      "detail": {
        "IncludeLabel": {
          "app": "monitor"
        }
      },
      "type": "service_docker_stdout"
    }
  ],
  "processors": [
    {
      "type": "processor_json",
      "detail": {
        "SourceKey": "content",
        "NoKeyError": true,
        "KeepSource": false
      }
    }
  ]
}
```

5.3.1.5.12. 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

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 Logtail 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
```


 **Notice** 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
<code>\${your_region_name}</code>	The region of the project. For more information, see View the information of a project .
<code>\${your_aliyun_user_id}</code>	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 an account ID for a server .
<code>\${your_machine_group_user_defined_id}</code>	The custom ID of your server group. For information about how to set the custom ID, see Step 1 in Create a machine group based on a custom ID .

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/log-service/logtail
```

Notice

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_CONFIG`.
- 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

- Log on to the [Log Service console](#).
- Click a project name.
- In the left-side navigation pane, click the **Server Groups** icon to show the server group list.
- 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.

- In the dialog box that appears, select **Custom ID** from the Identifier drop-down list. Enter the value of `ALIYUN_LOGTAIL_USER_DEFINED_ID` 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 heart beat 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 stdout and stderr logs from containers](#).
- [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 events, size:0
[2018-02-06 08:13:35.722155] [INFO] [8] [build/release64/sls/ilogtail/EventDispatcher.cpp:511] add existed check point events, size: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$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fbeb2bdb95d13b1e110172ef57
fe840c82155/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44
beab6e69718/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640
b1e16c22dbe/merged: must be superuser to unmount
...
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
ilogtail is running
```

- Restart Logtail.

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
```

5.3.1.6. Custom plug-ins

Context

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 in Linux](#).

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 [Configure data processing methods](#).

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.

5.3.1.6.1. Collect MySQL binary logs

This topic describes how to create a Logtail configuration in the Log Service console to collect MySQL binary logs.

Prerequisites

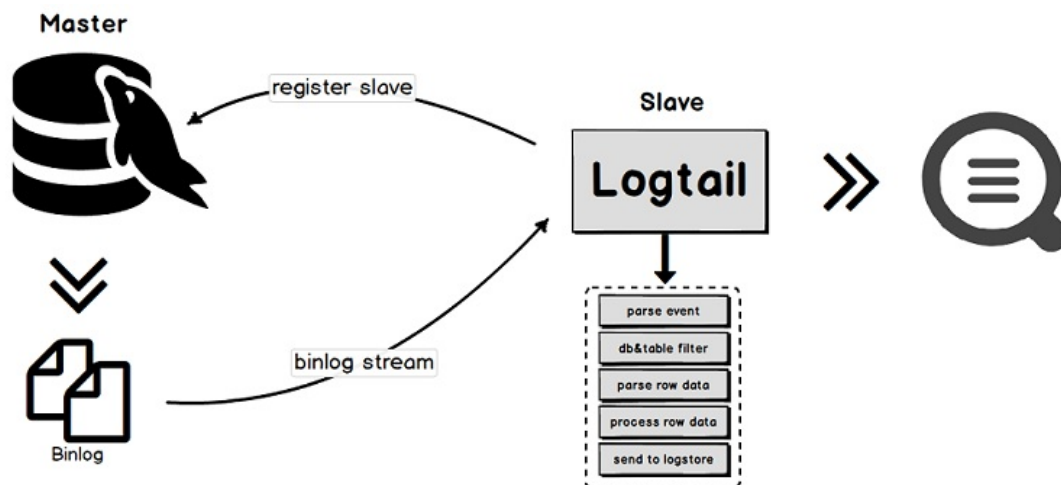
Logtail is installed on the server from which you want to collect MySQL binary logs. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).

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.
3. 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.

- 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:

```
show variables like "binlog_format";
```

In this example, the following output is returned:

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| 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 [Set Logtail startup parameters](#).

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 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 Log Service console](#).
2. In the **Import Data** section, select **MySQL BinLog - Plug-in**.
3. 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 **Next**.

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. 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 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**.

Notice 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. 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 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 [Configure data processing methods](#).

```
{
  "inputs": [
    {
      "type": "service_canal",
      "detail": {
        "Host": "*****.mysql.rds.aliyuncs.com",
        "Port": 3306,
        "User" : "root",
        "ServerID" : 56321,
        "Password": "*****",
        "IncludeTables": [
          "user_info\\.*"
        ],
        "ExcludeTables": [
          ".*\\.\\S+_inner"
        ],
        "TextToString" : true,
        "EnableDDL" : true
      }
    }
  ]
}
```

Parameter	Type	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: <pre>CREATE USER canal IDENTIFIED BY 'canal'; GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'canal'@'%'; -- GRANT ALL PRIVILEGES ON *.* TO 'canal'@'%'; FLUSH PRIVILEGES;</pre>

Parameter	Type	Required	Description
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 <code>/usr/local/ilogtail/user_log_config.json</code> 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.  Note The value of the ServerID parameter must be unique for your database. Otherwise, data collection fails.
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: <code>test_db.test_table</code>. 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 <code>.*\..*</code>.  Note To implement exact match, add <code>^</code> to the beginning of a regular expression and <code>\$</code> to the end. Example: <code>^test_db\\.test_table\$</code>.
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: <code>test_db.test_table</code>. 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.  Note To implement exact match, add <code>^</code> to the beginning of a regular expression and <code>\$</code> to the end. Example: <code>^test_db\\.test_table\$</code>.

Parameter	Type	Required	Description
StartBinName	string	No	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 the StartBinlogPos 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. <pre>show binary logs;</pre> Example: <pre>+-----+-----+ Log_name File_size +-----+-----+ mysql-bin.000063 241 mysql-bin.000064 241 mysql-bin.000065 241 mysql-bin.000066 10778 +-----+-----+ 4 rows in set (0.02 sec)</pre>  Note If you specify a value for the StartBinName parameter, a large volume of traffic is generated when Logtail collects data for the first time.
StartBinlogPos	int	No	The offset of the binary log file from which Logtail starts to collect data for the first time. Default value: 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.

Parameter	Type	Required	Description
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. Note If you set the value to true, the IncludeTables and ExcludeTables parameters become unavailable.
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. The old_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 false, the data on ROW_INSERT events contains the c1, c2, and c3 fields. 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. Note This parameter is available only for Logtail V0.16.19 and later.
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 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 [Enable the index feature and configure indexes for a Logstore](#).

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 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's 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 `time` (`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
_event_: row_update
_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.**.**
time: 2017-10-31 12:04:54
```

- DDL statement

```

_source_: 203.0.**.**
_tag_: _hostname_: i2bp145dd9fccu***
_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
<code>_host_</code>	The hostname of the database.
<code>_db_</code>	The name of the database.
<code>_table_</code>	The name of the table.
<code>_event_</code>	The type of the event.
<code>_id_</code>	The auto-increment ID. IDs start from 0 and increment by 1 each time data on a binary log event is collected.
<code>_gtid_</code>	The GTID.
<code>_filename_</code>	The name of the binary log file.
<code>_offset_</code>	The offset of the binary log file. The value is updated only when a COMMIT operation is performed.

5.3.1.6.2. Collect MySQL query results

This topic describes how to create a Logtail configuration in the Log Service console to collect MySQL query results.

Prerequisites

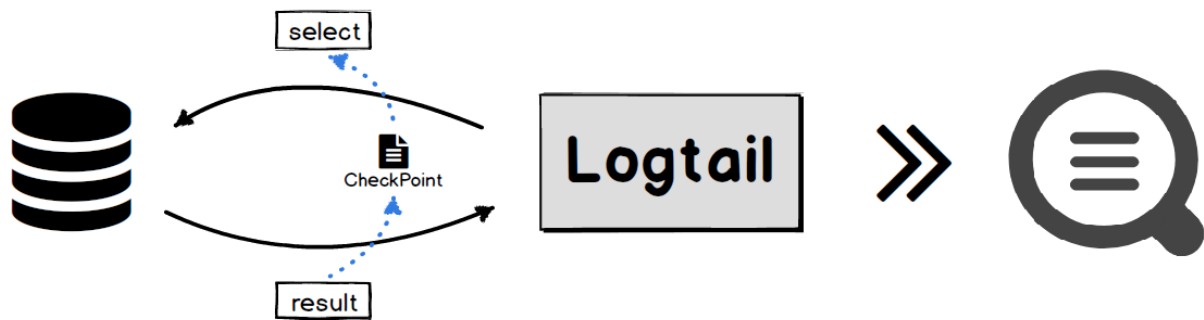
Logtail is installed on the server from which you want to collect MySQL query results. For more information, see [Install Logtail in Linux](#) or [Install Logtail in Windows](#).

 **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 Log Service console](#).

2. Select a data source.

In the **Import Data** section, select **MySQL Query Result - Plug-in**.

3. 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 **Next**.

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. 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 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**.

May Server Groups

Source Server Groups

Search by server group name

☒ ...

1 Items

Applied Server Groups

Search by server group name

0 Items

>

<

Previous

Next

Notice If you want to apply a machine group immediately after it is created, the heart beat 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**.

- 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 [Configure data processing methods](#).

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": {
        "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	Type	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 <code>/usr/local/ilogtail/user_log_config.json</code> 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.

Parameter	Type	Required	Description
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 <code>SELECT * from ... where id > ?</code> 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. If 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 the Limit 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: 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.
CheckPointColumnType	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.
CheckPointSavePerPage	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 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 [Enable the index feature and configure indexes for a Logstore](#).

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.

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.
2. Find the service_mysql keyword in the `/usr/local/ilogtail/user_log_config.json` file and modify parameters such as Address, User, and Password.
3. 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"
```

5.3.1.6.3. 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 in Linux](#) or [Install Logtail in Windows](#).

Note Linux servers support Logtail V0.16.13 or later. Windows servers support Logtail V1.0.0.8 or later.

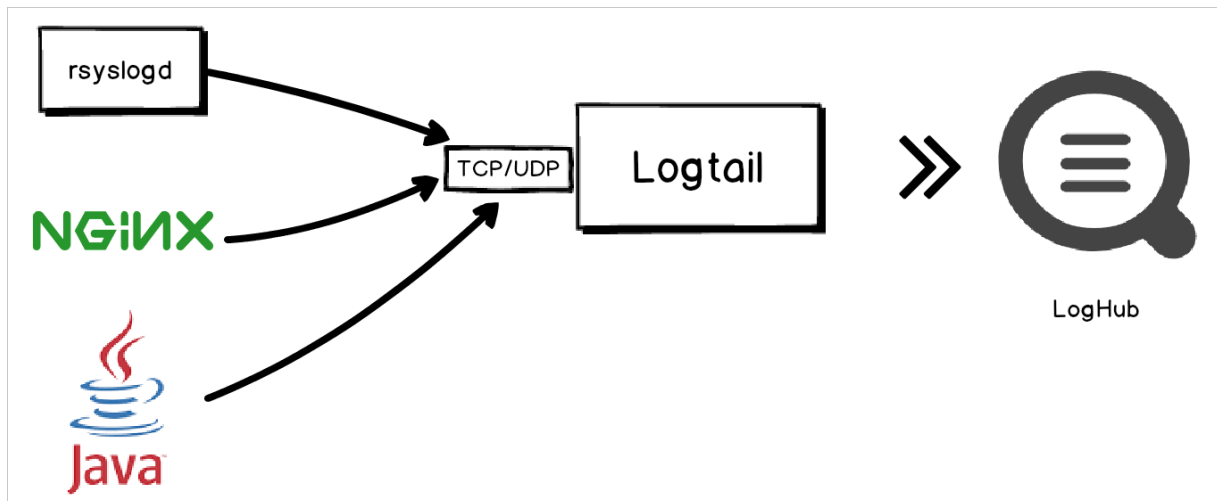
Overview

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 Log Service console](#).

3. In the **Import Data** section, select **Custom Data Plug-in**.

4. 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.

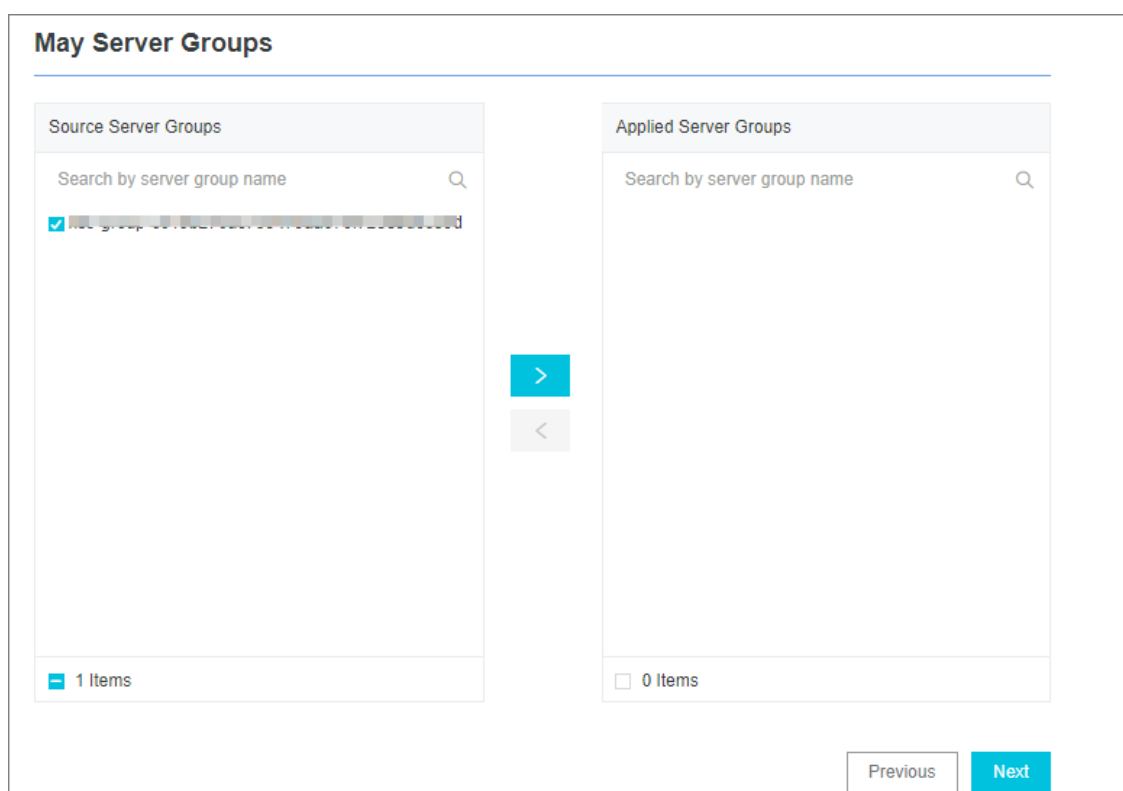
5. Create a machine group and click **Next**.

Before you can create a machine group, you must install Logtail.

Install Logtail as prompted. 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 a machine group is created, click **Use Existing Machine Groups** to select the machine group.

6. Select a machine group, move the machine group from **Source Machine Groups** to **Applied Server Groups**, and then click **Next**.



Notice 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**.

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.

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 [Configure data processing methods](#).

The following example shows how to configure Logtail plug-ins to listen on 127.0.0.1:9000 over UDP and TCP:

```
{
  "inputs": [
    {
      "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	Type	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]://[ip]:[port] 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 0.0.0.0. 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.

Parameter	Type	Required	Description
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 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 [Enable the index feature and configure indexes for a Logstore](#).

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.

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's next

After Logtail collects and sends syslogs to Log Service, you can view the logs in the Log Service console.

1	08-20 16:38:44	<pre> __source__: 127.0.0.1 __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 _unixtimestamp__: 1534754324 </pre>

Log field	Description
<code>_hostname_</code>	The hostname. If no hostname is included in the log, the hostname of the current host is obtained.
<code>_program_</code>	The tag field in the syslog protocol.
<code>_priority_</code>	The priority field in the syslog protocol.
<code>_facility_</code>	The facility field in the syslog protocol.
<code>_severity_</code>	The severity field in the syslog protocol.
<code>_unixtimestamp_</code>	The timestamp of the log.
<code>_content_</code>	The content of the log. If the log fails to be parsed, this field contains the content of the raw log.
<code>_ip_</code>	The IP address of the current host.

5.3.1.6.4. 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 [Set Logtail startup parameters](#).

- 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 `__tag__`, `__path__`, and `__topic__` 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, see Add log fields .
processor_drop	You can use the processor_drop plug-in to drop specified fields. For more information, see Drop log fields .
processor_rename	You can use the processor_rename plug-in to rename specified fields. For more information, see Rename 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) .

Logtail plug-in	Description
processor_json	You can use the processor_json plug-in to expand JSON fields. For more information, see Expand JSON fields .
processor_filter_regex	You can use the processor_filter_regex plug-in to filter logs. For more information, see Filter 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 `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	Type	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 <code>()</code> .
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.

Parameter	Type	Required	Description
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-j
ava"
```

- Logtail plug-in configurations for data processing

```
{
  "type" : "processor_regex",
  "detail" : {"SourceKey" : "content",
    "Regex" : "([\\d\\.]+) \\S+ \\S+ \\[[\\S+ \\S+\\] \\\"(\\w+) ([^\\\"]*)\\\" ([\\d\\.]+) (\\
d+) (\\d+) (\\d+|-) \\\"([\\\"]*)\\\" \\\"([\\\"]*)\\\" (\\d+)\"",
    "Keys" : ["ip", "time", "method", "url", "request_time", "request_length", "status", "
length", "ref_url", "browser"],
    "NoKeyError" : true,
    "NoMatchError" : true,
    "KeepSource" : false
  }
}
```

- Result

```
"ip" : "203.0.113.10"
"time" : "10/Aug/2017:14:57:51"
"method" : "POST"
"url" : "/PutData?Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%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	Type	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 Anchors parameter.

Parameter	Type	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.
ExpondConnector	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",
        "Stop" : "\t",
        "FieldName" : "time",
        "FieldType" : "string",
        "ExpondJson" : false
      },
      {
        "Start" : "json:",
        "Stop" : "",
        "FieldName" : "val",
        "FieldType" : "json",
        "ExpondJson" : true
      }
    ]
  }
}
```

- Result

```
"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	Type	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, <code>\u0001</code> .
SplitKeys	String array	Yes	The names of the delimited fields, for example, <code>["ip","time","method"]</code> .

Parameter	Type	Required	Description
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=YunOsAccountOpLog&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=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%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 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	Type	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, <code>\u0001\u0002</code> .
SplitKeys	String array	Yes	The names of the delimited fields, for example, <code>["key1", "key2"]</code> .
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: <code>false</code> . This value indicates that excess fields are not retained.
ExpandOthers	Boolean	No	Specifies whether to parse excess fields. Default value: <code>false</code> . This value indicates that excess fields are not parsed.
ExpandKeyPrefix	String	No	The name prefix of excess fields. For example, if you specify <code>expand_</code> for the parameter, the first two excess fields are named <code>expand_1</code> and <code>expand_2</code> .
NoKeyError	Boolean	No	Specifies whether to report an error if a field is not matched. Default value: <code>false</code> . This value indicates that no error is reported if a field is not matched.

Parameter	Type	Required	Description
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>&Date=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 `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_key_value`.

 **Note** Only Logtail V0.16.26 or later supports the plug-in.

Parameter	Type	Required	Description
SourceKey	String	Yes	The name of the source field.
Delimiter	String	No	The delimiter between key-value pairs. Default value: <code>\t</code> .
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.
DiscardWhenSeparatorNotFound	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 `processor_geoip`.

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	Type	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, <code>/user/data/GeoLite2-City_20180102/GeoLite2-City.mmdb</code> .
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_province_code_" : "ZJ"
"source_ip_country_code_" : "CN"
"source_ip_longitude_" : "120.*****"
"source_ip_latitude_" : "30.*****"
```

Filter logs by using regular expressions

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`.

 **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	Type	Required	Parameters
Include	JSON object that contains 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 contains 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	Type	Required	Description
Fields	Map	No	The key-value pairs that you want to add. You can specify multiple key-value pairs in the parameter.
IgnoreIfExist	Boolean	No	Specifies whether to retain key-value pairs that have the same key. Default value: <code>false</code> . 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`.

 **Note** Only Logtail V0.16.28 or later supports the plug-in.

Parameter	Type	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

```
{
  "processors": [
    {
      "type": "processor_drop",
      "detail": {
        "DropKeys": ["aaa1", "aaa2"]
      }
    }
  ]
}
```

- 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`.

 **Note** Only Logtail V0.16.28 or later supports the plug-in.

Parameter	Type	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.

Parameter	Type	Required	Description
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.
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.

- 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

```
{
  "processors":[
    {
      "type":"processor_gotime",
      "detail": {
        "SourceKey": "s_key",
        "SourceFormat":"2006-01-02 15:04:05",
        "SourceLocation":8,
        "DestKey":"d_key",
        "DestFormat":"2006/01/02 15:04:05",
        "DestLocation":9,
        "SetTime": true,
        "KeepSource": true,
        "NoKeyError": true,
        "AlarmIfFail": true
      }
    }
  ]
}
```

- 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`.

 **Note** Only Logtail V0.16.28 or later supports the plug-in.

Parameter	Type	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",
        "NoKeyError": true,
        "ExpandDepth": 0,
        "ExpandConnector": "-",
        "Prefix": "j",
        "KeepSource": false,
        "UseSourceKeyAsPrefix": true
      }
    }
  ]
}
```

- Result

```
"s_key": "{\\"k1\\": {\\"k2\\": {\\"k3\\": {\\"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 `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	Type	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.
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

The following example shows how to encapsulate the a and b fields into the d_key field.

- Raw log

```
"a": "1"
"b": "2"
```

- Logtail plug-in configurations for data processing

```
{
  "processors": [
    {
      "type": "processor_packjson",
      "detail": {
        "SourceKeys": ["a", "b"],
        "DestKey": "d_key",
        "KeepSource": true,
        "AlarmIfEmpty": true
      }
    }
  ]
}
```

- Result

```
"a": "1"
"b": "2"
"d_key": "{\"a\": \"1\", \"b\": \"2\"}"
```

Rename log fields

You can use this method to rename multiple fields.

The type of the plug-in is `processor_rename`.

- 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	Type	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
      }
    }
  ]
}
```

- Result

```
"bbb1": "value1"
"bbb2": "value2"
"aaa3": "value3"
```

Extract log time (strftime)

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 `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 Logtail V0.16.28 or later supports the plug-in.

Parameter	Type	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 format. The current time zone of the server where Logtail is installed is used.

```
%Y/%m/%d %H:%M:%S
```

- 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
```

- 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,requ
estTime=1508729889913,latency=22ms,
request={appKey:ba,optional:{\\domains\\:\\daily\\,\\version\\:\\v2\\},rawQuery:{\\query\\:\\The ro
ute to Location A\\,\\domain\\:\\Navigation\\,\\intent\\:\\navigate\\,\\slots\\:\\to_geo:level3=Loc
ation 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"
"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\\:\\\\navigate\\,\\slots\\:\\to_geo:level3=\\344\\271\\220\\345\\261\\261\\,\\location\\:\\\\345\\214\\227\\344\\272\\254\\},requestId:52460dbed4d540b88a973cf5452b1447}"
"response_answers" : "[]"
"response_status" : "SUCCESS"

```

5.3.1.7. 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 <code>.log*</code> and <code>.log</code> 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 excess 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.

Item	Description
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 <code>'\0'</code> in multiple lines, the log is truncated at the first <code>'\0'</code> .

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 <code>/tmp/logtail_checkpoint</code> directory. You can modify the values of the related parameters. For more information, see Set Logtail startup parameters .

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 Set Logtail startup parameters .
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.

Item	Description
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 <code>mem_usage_limit</code> parameter to raise the threshold of memory resources that are available for Logtail. For more information, see Set Logtail startup parameters. 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 Set Logtail startup parameters .
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.

5.3.2. Other collection methods

5.3.2.1. 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.

Context

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  > **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));
            // 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 <i>Log Service Developer Guide</i> .

Parameter	Required	Description
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 SlsWebLogger.

```
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 <i>Log Service Developer Guide</i> .
\${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.

```
<img src='http://${project}.${host}/logstores/${logstore}/track.gif?APIVersion=0.6.0&key1=val1&key2=val2' />
<img src='http://${project}.${host}/logstores/${logstore}/track_ua.gif?APIVersion=0.6.0&key1=val1&key2=val2' />
```

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*.

5.3.2.2. Collect logs over the syslog protocol

This topic describes how to collect logs from a server over the syslog protocol. If you use this method, you do not need to use third-party agents to forward collected logs.

Limits

- Syslogs must conform to RFC 5424.
- The maximum size of each log is 64 KB.
- TCP-based Transport Layer Security (TLS) 1.2 must be used for data transmission.

Procedure

To upload data over the syslog protocol, associate the protocol with the endpoint of Log Service. Set the port number of the syslog protocol to 10009. Select the endpoint of your project. Specify a project, Logstore, and AccessKey pair for Log Service in the STRUCTURED-DATA field of syslogs. The following table describes the parameters.

Parameter	Description	Example
STRUCTURED-DATA	Each STRUCTURED-DATA field must be named Logservice.	Logservice
Project	A project in Log Service. Create a project in Log Service before you collect data.	test-project-1
Logstore	The name of the Logstore. You must create a Logstore in Log Service before you collect logs.	test-logstore-1
access-key-id	Your AccessKey ID.	<yourAccessKeyId>
access-key-secret	Your AccessKey secret.	<yourAccessKeySecret>

Sample log

Log Service parses received syslogs. During the parsing process, Log Service deletes the Logservice field to ensure the security of the AccessKey pair. The following table describes fields that are received by Log Service.

Field	Description
__source__	The HOSTNAME field in the syslog.
__topic__	The value of this field is syslog-forwarder.
__facility__	The facility (device or module) information in the syslog.
__program__	The name of the process.

Field	Description
<code>__severity__</code>	The severity of the syslog.
<code>__priority__</code>	The priority of the syslog.
<code>__unixtimestamp__</code>	The timestamp of the syslog. Unit: nanoseconds.
<code>content</code>	The MSG field in the syslog.

For more information, see RFC 5424.

Examples

You want to collect syslogs from your host and write them into Log Service. The project in Log Service is named `test-project-1` and the Logstore is named `test-logstore-1`. The project resides in the China (Qingdao) region (cn-qingdao-env12-d01). The AccessKey ID of the RAM user with the write permission is `<yourAccessKeyId>`, and the AccessKey secret is `<yourAccessKeySecret>`.

- Example 1: Use rsyslog to forward syslogs to Log Service

Rsyslog is installed on Linux hosts by default to process syslogs. You can configure rsyslog to forward syslogs to Log Service. The configuration file of rsyslog varies based on rsyslog versions. You can run the `man rsyslogd` command to view the rsyslog version.

Note The rsyslog-gnutls package must be installed. To install the package, you can run the `sudo apt-get install rsyslog-gnutls` or `sudo yum install rsyslog-gnutls` command.

- Select either of the following code samples based on your rsyslog version and add the code to the end of the rsyslog configuration file. Replace the parameter values. The directory of the rsyslog configuration file is `/etc/rsyslog.conf`.

- Rsyslog v8 and later

Set `$DefaultNetstreamDriverCAFile` to the location of the root certificate in the system.

```
# Setup disk assisted queues
$WorkDirectory /var/spool/rsyslog # where to place spool files
$ActionQueueFileName fwdRule1    # unique name prefix for spool files
$ActionQueueMaxDiskSpace 1g       # 1gb space limit (use as much as possible)
$ActionQueueSaveOnShutdown on     # save messages to disk on shutdown
$ActionQueueType LinkedList       # run asynchronously
$ActionResumeRetryCount -1        # infinite retries if host is down
$ActionSendTCPrebindInterval 100  # close and re-open the connection to the remote host every
100 of messages sent.
#RsyslogGnuTLS set to default ca path
$DefaultNetstreamDriverCAFile /etc/ssl/certs/ca-bundle.crt
template(name="LogServiceFormat" type="string"
string="<%pri%>1 %timestamp:::date-rfc3339% %HOSTNAME% %app-name% %procid% %msgid% [logservice project=\"test-project-1\" logstore=\"test-logstore-1\" access-key-id=\"<yourAccessKeyId>\" access-key-secret=\"<yourAccessKeySecret>\"] %msg%\n"
)
# Send messages to Loggly over TCP using the template.
action(type="omfwd" protocol="tcp" target="data.cn-qingdao-env12-d01.sls-pub.cloud.env12.shuguang.com" port="10009" template="LogServiceFormat" StreamDriver="gtls" StreamDriverMode="1" StreamDriverAuthMode="x509/name" StreamDriverPermittedPeers="*.log.aliyuncs.com")
```

■ Rsyslog v7 and earlier

Set `$DefaultNetstreamDriverCAFile` to the location of the root certificate in the system.

```
# Setup disk assisted queues
$WorkDirectory /var/spool/rsyslog      # where to place spool files
$ActionQueueFileName fwdRule1         # unique name prefix for spool files $ActionQueueMaxDis
iskSpace 1g                          # 1gb space limit (use as much as possible) $ActionQueueSaveOnShutdow
n on                                # save messages to disk on shutdown
$ActionQueueType LinkedList           # run asynchronously
$ActionResumeRetryCount -1            # infinite retries if host is down $ActionSendTCPRebi
ndInterval 100                      # close and re-open the connection to the remote host every 100 of mess
ages sent.
# RsyslogGnuTLS set to default ca path
$DefaultNetstreamDriverCAFile /etc/ssl/certs/ca-bundle.crt
$ActionSendStreamDriver gtls
$ActionSendStreamDriverMode 1
$ActionSendStreamDriverAuthMode x509/name
$ActionSendStreamDriverPermittedPeer data.cn-qingdao-env12-d01.sls-pub.cloud.env12.shuguang.c
om
template(name="LogServiceFormat" type="string" string="<%pri%>1 %timestamp:::date-rfc3339% %H
OSTNAME% %app-name% %procid% %msgid% [logservice project=\"%test-project-1\" logstore=\"%test-l
ogstore-1\" access-key-id=\"%yourAccessKeyId%\" access-key-secret=\"%yourAccessKeySecret%\"]
%msg%\n")
*. * action(type="omfwd" protocol="tcp" target="data.cn-qingdao-env12-d01.sls-pub.cloud.env12.
shuguang.com" port="10009" template="LogServiceFormat")
```

ii. Run the `sudo service rsyslog restart` , `sudo /etc/init.d/syslog-ng restart` , or `systemctl restart rsyslog` command to restart rsyslog.

iii. If no syslog is generated, you can run the `logger` command to generate several test logs, for example, `logger hello world!` .

● Example 2: Use syslog-ng to forward syslogs to Log Service

syslog-ng is an open-source log management daemon that implements the syslog protocol for Unix and Unix-like operating systems. You can run the `sudo yum install syslog-ng` or `sudo apt-get install syslog-ng` command to install syslog-ng.

```
### Syslog-ng Logging Config for LogService ###
template LogServiceFormat {
    template("<${PRI}>1 ${ISODATE} ${HOST:--} ${PROGRAM:--} ${PID:--} ${MSGID:--} [logservice proje
ct=\"%test-project-1\" logstore=\"%test-logstore-1\" access-key-id=\"%yourAccessKeyId%\" access-key-s
ecret=\"%yourAccessKeySecret%\"] $MSG\n"); template_escape(no);
};
destination d_logservice{
    tcp("data.cn-qingdao-env12-d01.sls-pub.cloud.env12.shuguang.com" port(10009)
    tls(peer-verify(required-untrusted))
    template(LogServiceFormat));
};
log {
    source(s_src); # default use s_src
    destination(d_logservice);
};
### END Syslog-ng Logging Config for LogService ###
```

 **Note** By default, rsyslog is installed on each server to take over Syslog. Before you can use syslog-ng, you must uninstall rsyslog because rsyslog and syslog-ng cannot operate at the same time.

i. Replace the parameter values in the preceding sample code and add the code to the end of the syslog-ng

configuration file. The directory of the syslog-ng configuration file is `/etc/syslog-ng/syslog-ng.conf`.

- ii. Run the `sudo /etc/init.d/syslog-ng restart` , `sudo service syslog-ng restart` , or `sudo systemctl restart syslog-ng` command to restart syslog-ng.
- iii. If no syslog is generated, you can run the `logger` command to generate several test logs, for example, `logger hello world!` .

FAQ

- How can I send syslogs to Log Service?

You can use Ncat commands to send syslogs to Log Service. The command checks whether the network connection is normal and whether the AccessKey pair is authorized to send syslogs. If Ncat is not installed on your host, you can run the `sudo yum install nmap-ncat` command to install Ncat. For example, you can run the following command to send a syslog to Log Service. The project in Log Service is named `test-project-1` and the Logstore is named `test-logstore-1` . The project resides in the China (Qingdao) region (cn-qingdao-env12-d01). The AccessKey ID of the RAM user with the write permission is `<yourAccessKeyId>` , and the AccessKey secret is `<yourAccessKeySecret>` .

Note

- The timestamp of the syslog uses UTC+0, for example, 2019-03-28T03:00:15.003Z. In UTC+8, this time is 2019-03-28T11:00:15.003.
- Ncat commands cannot automatically check whether the network is disconnected. You must enter a syslog and press Enter to send the syslog within 30 seconds after you run the Ncat command.

```
[root@i2bpl45dd9fccuid7g**** ~]# ncat --ssl data.cn-qingdao-env12-d01.sls-pub.cloud.env12.shuguang.com 10009
<34>1 2019-03-28T03:00:15.003Z mymachine.example.com su - ID47 [logservice project="test-project-1" logstore="test-logstore-1" access-key-id="<yourAccessKeyId>" access-key-secret="<yourAccessKeySecret>"] this is a test message
```

After you send the syslog, you can preview the syslog in the Log Service console. For more information, see [Preview log data in the Log Service console](#).

- How can I view rsyslog error messages?

You can run the vim command to view rsyslog error messages, which are stored in the `/var/log/message` directory by default.

- Rsyslog error message example 1

```
dlopen: /usr/lib64/rsyslog/lmnsd_gtls.so: cannot open shared object file: No such file or directory
```

This error message returns if the rsyslog-gnutls package is not installed. You can run the `sudo apt-get install rsyslog-gnutls` or `sudo yum install rsyslog-gnutls` command to install rsyslog, and then restart rsyslog.

- Rsyslog error message example 2

```
unexpected GnuTLS error -53 - this could be caused by a broken connection. GnuTLS reports:Error in the push function
```

This error message returns if the TCP connection is ended due to a long period of inactivity. You can ignore this message because rsyslog reestablishes the connection.

- How can I view rsyslog-ng error messages?

You can run the `systemctl status syslog-ng.service` and `journalctl-xe` commands to view rsyslog-ng error messages, which are stored in journals by default.

The following example shows the error message that returns if syslog-ng fails to start:

```
Job for syslog-ng.service failed because the control process exited with error code. See "systemctl status syslog-ng.service" and "journalctl -xe" for details
```

If this error message occurs, check whether the configuration file format is valid or whether configuration conflicts exist. For example, you cannot configure multiple `internal()` sources.

5.3.2.3. Use SDKs to collect logs

5.3.2.3.1. 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](#).

5.3.2.3.2. 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](#).

5.3.2.3.3. 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](#).

5.3.2.3.4. 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](#).

5.3.2.3.5. 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_PROJECT', ' '), "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 `aliyun.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 [python_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, <code>sls</code> 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.

Field	Description
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 `fields` parameter of a class. For more information, see [aliyun.log.LogFields](#).

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.

```
#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 `aliyun.log.QueuedLogHandler` class, pass named parameters to the constructor. For more information, see [Parameters](#).

5.3.2.4. Collect common logs

5.3.2.4.1. 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>
  <Loggers>
    <Logger name="com.foo.Bar" level="trace">
      <AppenderRef ref="Console"/>
    </Logger>
    <Root level="error">
      <AppenderRef ref="Console"/>
    </Root>
  </Loggers>
</Configuration>
```

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 timeo
ut,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+,\d+)\s\[([^\]]*)\]\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.

Key	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 [Python logs](#).

5.3.2.4.2. 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.

Context

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.

```
import logging
import logging.handlers
LOG_FILE = 'tst.log'
handler = logging.handlers.RotatingFileHandler(LOG_FILE, maxBytes = 1024*1024, backupCount = 5) # Create a handler object.
fmt = '%(asctime)s - %(filename)s:%(lineno)s - %(levelname)s %(levelname)s %(pathname)s %(module)s %(funcName)s %(created)f %(thread)d %(threadName)s %(process)d %(name)s - %(message)s' // Define the output format of logs.
formatter = logging.Formatter(fmt) # Create a formatter object.
handler.setFormatter(formatter) # Add the formatter to the handler.
logger = logging.getLogger('tst') # Retrieve a logger that is named tst.
logger.addHandler(handler) # Add the handler to the logger.
logger.setLevel(logging.DEBUG)
logger.info('first info message')
logger.debug('first debug message')
```

The following table describes the fields in the formatter configurations.

Field	Description
<code>%(name)s</code>	The name of the logger that generates a log.
<code>%(levelname)s</code>	The severity level of a log in the numeric format.
<code>%(levelname)s</code>	The severity level of a log in the text format. Valid values: DEBUG, INFO, WARNING, ERROR, and CRITICAL.
<code>%(pathname)s</code>	The full path name of the source file where the logging call is initiated.
<code>%(filename)s</code>	The name of the source file.
<code>%(module)s</code>	The name of the module where the logging call is initiated.

Field	Description
%(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**.
3. 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 **Next**.
Before you can create a machine group, you must install Logtail.
Install Logtail as prompted. 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 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**.

The screenshot shows a web interface titled "May Server Groups". It features two main panels: "Source Server Groups" on the left and "Applied Server Groups" on the right. Both panels have a search bar labeled "Search by server group name". In the "Source" panel, one item is listed and selected with a checkmark. In the "Applied" panel, no items are listed. Between the panels are navigation arrows: a blue right arrow and a grey left arrow. At the bottom of each panel, it says "1 Items" and "0 Items" respectively. At the bottom right of the interface are "Previous" and "Next" buttons.

Notice 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. 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 **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) 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.  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 <code>__raw__</code> 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 the Custom 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 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.

Parameter	Description
Filter Configuration	Only logs that meet all filter conditions are collected. Examples: - Collect logs that meet specified conditions: If you set Key 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 .*^(?!.*(healthcheck)).*, 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 [Enable the index feature and configure indexes for a Logstore](#).

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 Python logs.

5.3.2.4.3. 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'
    }
  ]
});
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 [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

```
\[([^\]]+)\]\s\((\w+)\)\s(\w+)\s-\s(\S+)\s-\s-\s"([^\"]+)"\s\((\d+)\s[^\s]+)\s"([^\"]+)"\s"([^\"]+)"\s.*
```

- Extracted fields

```
time , level , loggerName , ip, request , status , referer , and user_agent
```

5.3.2.4.4. 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

```
(\S+) - - \[([^\]]*)] "(\\S+) ([^"]+)" (\\S+) (\\S+) "([^\"]+)" "([^\"]+)"
```

- Time conversion format

```
%d/%b/%Y:%H:%M:%S
```

- The following table lists the extraction results of the sample log.

Key	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

5.3.2.4.5. ThinkPHP logs

ThinkPHP is a web application development framework based on PHP.

Log format

ThinkPHP logs are displayed in the following format:

```
<? php
Think\Log::record('Failed to find a model class that you can use the D method to instantiate');
? >
```

Sample logs

```
[ 2016-05-11T21:03:05+08:00 ] 10.10.10.1 /index.php
INFO: [ app_init ] --START--
INFO: Run Behavior\BuildLiteBehavior [ RunTime:0.000014s ]
INFO: [ app_init ] --END-- [ RunTime:0.000091s ]
INFO: [ app_begin ] --START--
INFO: Run Behavior\ReadHtmlCacheBehavior [ RunTime:0.000038s ]
INFO: [ app_begin ] --END-- [ RunTime:0.000076s ]
INFO: [ view_parse ] --START--
INFO: Run Behavior\ParseTemplateBehavior [ RunTime:0.000068s ]
INFO: [ view_parse ] --END-- [ RunTime:0.000104s ]
INFO: [ view_filter ] --START--
INFO: Run Behavior\WriteHtmlCacheBehavior [ RunTime:0.000032s ]
INFO: [ view_filter ] --END-- [ RunTime:0.000062s ]
INFO: [ app_end ] --START--
INFO: Run Behavior\ShowPageTraceBehavior [ RunTime:0.000032s ]
INFO: [ app_end ] --END-- [ RunTime:0.000070s ]
ERR: Failed to find a model class that you can use the D method to instantiate
```

Configure Logtail to collect ThinkPHP logs

The automatically generated regular expression is based on the sample log and may not apply to other logs. Therefore, you must make minor changes to the regular expression after it is generated.

Each ThinkPHP log contains multiple lines and variable information. You can extract the time, IP address, URL, and message fields from ThinkPHP logs. The message field contains multi-line information because such information is variable and must be included into one field.

Configurations required for collecting ThinkPHP logs:

- Regular expression that matches the first line of a multi-line log

```
\[\s\d+-\d+-\w+:\d+:\d+\+\d+:\d+\s. *
```

- Regular expression

```
\[\s(\d+-\d+-\w+:\d+:\d+)[^:]+\d+\s]\s+(\S+)\s(\S+)\s+(\. *)
```

- Time expression

```
%Y-%m-%dT%H:%M:%S
```

5.3.2.4.6. Collect Unity3D logs

This topic describes how to use the web tracking feature of Log Service to collect Unity3D logs.

Context

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 [WebTracking](#).

2. 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)
    {
        WWW 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 Unity3D 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");
    }
}
```

4. 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.

5.4. Query and analysis

5.4.1. Log search overview

Log Service allows you to search for 1 billion to hundreds of billions of rows of log data within seconds. This topic describes the syntax and limits of the log search feature and provides examples.

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 (|). For more information about the query statement, see [Search syntax](#).

Note

- A search statement can be executed alone. However, an analytic statement must be executed together with a search statement. The log analysis feature is based on search results or all data in a Logstore.
- If you need to search for tens of billions of rows of data, 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 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. For more information, see Log analysis overview.

- Example

```
* | SELECT status, count(*) AS PV GROUP BY status
```

Limits

- Each project supports a maximum of 1,000 concurrent search statements at the same time.
For example, 1,000 users can concurrently search for data in all Logstores of a project at the same time.
- You can specify a maximum of 30 keywords for each search statement.
- The maximum size of a field value is 10 KB. If the size of a field value exceeds 10 KB, the excess content is not queried.
- The returned logs are displayed on multiple pages. Each page displays a maximum of 100 search results.
- Log Service performs the DOM operation only on the first 10,000 characters of a log.
- If you perform a fuzzy search, Log Service searches for 100 words that meet the specified conditions. Logs that contain one or more of the 100 words and meet the search conditions are returned.

Search methods

 **Notice** Before you search for logs, you must make sure that logs are collected and indexes are configured for the fields. Indexes are used in a storage structure to sort one or more columns of log data. For more information, see [Enable the indexing feature and configure indexes for a Logstore](#).

- Use the Log Service console

Log on to the Log Service console. On the Search & Analysis page of a Logstore, specify a time range and execute a search statement. For more information, see [Query logs](#) and [Search syntax](#).

- Call API operations

Call the GetLogs and GetHistograms operations to search for log data. For more information, see the GetLogs and GetHistograms topics of API Reference in *Developer Guide*.

5.4.2. 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 [Enable the indexing feature and configure indexes for a Logstore](#). 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.

Statement	Description
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

```
* | 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](#)
 - [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](#)
 - [Binary string functions](#)
 - [Bitwise operations](#)
 - [Interval-valued comparison and periodicity-valued comparison functions](#)
 - [Comparison functions and operators](#)
 - [Lambda functions](#)
 - [Logical functions](#)

- [Geospatial functions](#)
- [Geography functions](#)
- [Machine learning syntax and functions](#)
- [GROUP BY syntax](#)
- [Window functions](#)
- [HAVING syntax](#)
- [ORDER BY syntax](#)
- [LIMIT syntax](#)
- [Syntax for CASE statements and if\(\) functions](#)
- [UNNEST function](#)
- [Field aliases](#)
- [Nested subqueries](#)

5.4.3. Configure indexes

Indexes are used in a storage structure 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](#).

Index types

The following table describes the index types that are 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 <code>error</code> returns the logs that contain the keyword <code>error</code> .
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 <code>level:error</code> returns the logs in which the value of the <code>level</code> field contains <code>error</code>. If you want to use the analysis feature, you must configure field indexes and turn on Enable Analytics for the required fields. The analysis feature does not generate index traffic or occupy storage space.

Notice

- The indexing feature is applicable only to the log data that is written to the current Logstore 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 Log Service console](#).
2. In the Projects section, click the project in which you want to query and analyze logs.

3. Choose **Log Storage > Logstores**. On the Logstores tab, click the Logstore in which logs are stored.
4. On the Search & Analysis page of the Logstore, choose **Index Attributes > Attributes**.
If the indexing feature is not enabled, click **Enable**.
5. Configure indexes in the Search & Analysis panel.

Parameter	Description
LogReduce	If you turn on LogReduce , Log Service automatically clusters text logs that have the same pattern during log collection. This way, you can obtain the overall information of the 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 <code>internalError</code>, you can search for the log only by using the keyword <code>internalError</code>. ◦ If you turn off Case Sensitive, searches are not case-sensitive. For example, if a log contains <code>internalError</code>, you can search for the log by using the keyword <code>INTERNALERROR</code> or <code>internalerror</code>.
Include Chinese	Specifies whether to distinguish between Chinese content and English content in searches. ◦ If you turn on Include Chinese and a log contains Chinese characters, the Chinese content is split based on the Chinese grammar. The English content is split based on specified delimiters.  Notice When the Chinese content is split, the data write speed is reduced. Proceed with caution. ◦ If you turn off Include Chinese, all content is split based on specified delimiters.
Delimiter	The delimiters that are used to split the content of a log into multiple words. The following delimiters are supported: <code>, '";=()[]{}?@&<>/: \n\t\r . \n</code> indicates a line feed, <code>\t</code> indicates a tab character, and <code>\r</code> indicates a carriage return. For example, the content of a log is <code>/url/pic/abc.gif</code> . ◦ If you do not specify a delimiter, the log is processed as a single word <code>/url/pic/abc.gif</code>. You can search for the log only by using the keyword <code>/url/pic/abc.gif</code>. You can also perform a fuzzy search by using the keyword <code>/url/pic/*</code>. ◦ If you set the delimiter to a forward slash (/), the content of the log is split into the following three words: <code>url</code>, <code>pic</code>, and <code>abc.gif</code>. You can search for the log by using the keyword <code>url</code>, <code>abc.gif</code>, or <code>/url/pic/abc.gif</code>. You can also perform a fuzzy search by using the keyword <code>pi*</code>. ◦ If you set the delimiter to a forward slash (/) and a period (.), the content of the log is split into the following four words: <code>url</code>, <code>pic</code>, <code>abc</code>, and <code>gif</code>.

6. Click **OK**.
The index configurations take effect within 1 minute.

Configure field indexes

1. [Log on to the Log Service console](#).

2. In the **Projects** section, click the project in which you want to query and analyze logs.
3. Choose **Log Storage > Logstores**. On the **Logstores** tab, click the Logstore in which logs are stored.
4. On the **Search & Analysis** page of the Logstore, choose **Index Attributes > Attributes**.
If the indexing feature is not enabled, click **Enable**.
5. Configure indexes in the **Search & Analysis** panel.

Parameter	Description
Key Name	The name of the log field. Example: <code>client_ip</code>.  Note ◦ When you configure an index for a tag field, you must specify the value of the Key Name parameter in the <code>__tag__:KEY</code> format. For example, you can set this parameter to <code>__tag__:__receive_time__</code>. Different tag fields are supported. For example, a tag field can indicate a public IP address or a UNIX timestamp. ◦ When you configure an index for a tag field, you must set the Type parameter for each tag field to text. Numeric data types are not supported.
Type	The data type of the log field value. Valid values: text, long, double, and json. For more information, see Data types.  Note If a field is of the long or double type, you cannot set the Case Sensitive, Include Chinese, or Delimiter parameter.
Alias	The alias of the field. Example: <code>ip</code>. An alias is used only in analytic statements. You must use the original field name in search statements. For more information, see Field 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 <code>internalError</code>, you can search for the log only by using the keyword <code>internalError</code>. ◦ If you turn off Case Sensitive, searches are not case-sensitive. For example, if a log contains <code>internalError</code>, you can search for the log by using the keyword <code>INTERNALERROR</code> or <code>internalerror</code>.
Delimiter	The delimiters that are used to split the content of a log into multiple words. The following delimiters are supported: <code>, '";=() [] {} ?@&<>/: \n \t \r . \n</code>. <code>\n</code> indicates a line feed, <code>\t</code> indicates a tab character, and <code>\r</code> indicates a carriage return. For example, the content of a log is <code>/url/pic/abc.gif</code>.

Parameter	Description
Include Chinese	Specifies whether to distinguish between Chinese content and English content in searches. - If you turn on Include Chinese and a log contains Chinese characters, the Chinese content is split based on the Chinese grammar. The English content is split based on specified delimiters.  Notice When the Chinese content is split, the data write speed is reduced. Proceed with caution. - If you turn off Include Chinese, all content is split based on specified delimiters.
Enable Analytics	To use the analysis feature, you must turn on Enable Analytics .

6. Click **OK**.

The index configurations take effect within 1 minute.

5.4.4. 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 [Enable the indexing feature and configure indexes for a Logstore](#).

Query and analyze logs

- [Log on to the Log Service console](#).
- In the **Projects** section, click the project to which the Logstore belongs.
- Choose **Log Storage > Logstores**. On the Logstores tab, click the Logstore where logs are stored.
- 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](#).

- 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.

- 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](#).

 **Note** The contextual query feature supports only the log data that is collected by Logtail.

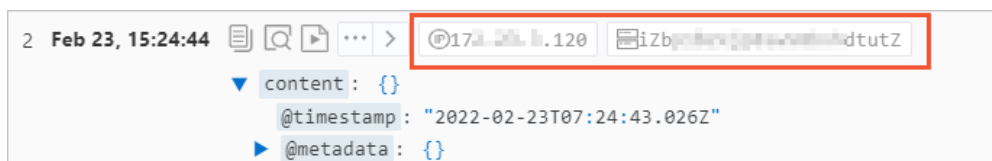
- LiveTail: If you click the  icon of a log on the **Raw Log** tab, you can monitor logs in real time and extract important information. For more information, see [LiveTail](#).

 **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 [Export logs](#).
- Column settings: You can click the  icon 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** To view the log content on the tab, select **Content**.

- JSON configurations: You can click the  icon and select **JSON Configurations** to 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.



• 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 [Drill-down analysis](#).
- 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](#).

5.4.5. Download logs

This topic describes how to download logs from Log Service to an on-premises host.

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  icon next to the name of the Logstore whose logs you want to download and select **Search & Analysis**.
4. In the upper-right corner of the **Raw Logs** tab, click the  icon.
5. In the **Log Download** dialog box, select a method that you can use to download logs and click **OK**.
 - **Download Log in Current Page**: Download the logs that are displayed on the current page to a file in the comma-separated values (CSV) format.
 - **Download All Logs Using Command Line Tool**: Download all logs as prompted.

5.4.6. Index data type

5.4.6.1. 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.	<code>uri:"login*" and method:"post"</code>
	long	The numeric type. You can specify numeric ranges to query indexes of this type.	<code>status in [200, 500]</code>
	double	The floating-point type.	<code>price>28.95</code>

Query type	Index data type	Description	Example
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.	<code>level0.key>29.95 and level0.key2:"action"</code>
	text	Creates indexes for all fields in a log except the time field. The data type of the indexes is text.	<code>error and "login fail"</code>

5.4.6.2. 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. 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.

5.4.6.3. 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.

Notice

- 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. -1000000 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	...	@17.1.1.78	1612236963	nginx_access_log
					__tag__:__client_ip__:47.1.1.166
					body_bytes_sent:2636
					client_ip:17.1.1.59
					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:119.1.1.54
					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

Key Name	Enable Search				Include Chinese	Enable Analytics	Delete
	Type	Alias	Case Sensitive	Delimiter: ?			
body_bytes_sent	long					☒	☐
client_ip	text		☐	','=000?@&<>:/\n\r	☐	☒	☐
host	text		☐	','=000?@&<>:/\n\r	☐	☒	☐
http_user_agent	text		☐	','=000?@&<>:/\n\r	☐	☒	☐
region	text		☐	','=000?@&<>:/\n\r	☐	☒	☐
remote_addr	text		☐	','=000?@&<>:/\n\r	☐	☒	☐
remote_user	text		☐	','=000?@&<>:/\n\r	☐	☒	☐
request_length	long					☒	☐
request_method	text		☐	','=000?@&<>:/\n\r	☐	☒	☐
request_time	long					☒	☐
request_uri	text		☐	','=000?@&<>:/\n\r	☐	☒	☐
status	long					☒	☐

- Query statements

- 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
```

- To query the log entries in which the response status code is 200, execute the following search statement:

```
status = 200
```

5.4.6.4. 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 `content.remote_addr`, `content.request.request_length`, and `content.request.request_method` fields.

```
content: {
  remote_addr:"192.0.2.0"
  request: {
    request_length:"73"
    request_method:"GE
```

Notice

- 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:

```

0. time:2018-01-01 12:00:00
1. class:central-log
2. status:200
3. latency:68.75
4. message:
  {
    "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.

Index configurations

Key Name	Type	Alias	Case Sensitive	Delimiter: ?	Include Chinese	Enable Analytics	Delete
class	text		☐	, "=000?@&<>\/\n\r	☐	☒	×
info	json		☐	, "=000?@&<>\/\n\r	☐	☐	×
methodName	text					☒	×
param.projectName	text					☒	×
param.requestId	text					☒	×
result.code	long					☒	×
result.message	text					☒	×
success	text					☒	×
usedTime	long					☒	×
latency	long					☒	×
status	long					☒	×

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"
```

5.4.7. Query syntax and functions

5.4.7.1. 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 `owner_id>100` 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.	<code>PUT and cn-shanghai</code> : returns the logs that contain the keywords <code>PUT</code> and <code>cn-shanghai</code> .
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 .	<code>request_time>60 and request_method:Ge*</code> : returns the logs in which the value of the <code>request_time</code> field is greater than 60 and the value of the <code>request_method</code> field starts with <code>Ge</code> .

- Exact searches and fuzzy searches

Search type	Description	Example
Exact search	Complete strings are used for queries.	◦ <code>host:www.yl.mock.com</code> : returns the logs in which the value of the <code>host</code> field is <code>www.yl.mock.com</code>. ◦ <code>PUT</code> : returns the logs that contain the keyword <code>PUT</code>.

Search type	Description	Example
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]. A fuzzy search is performed based on samples by using the following mechanism: ◦ 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 and returns part of the search results.	◦ <code>addr*</code> : searches for 100 strings that start with addr from logs, and returns the logs that contain one or more of these strings. ◦ <code>host:www.yl*</code> : 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.

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:
 - Colons (:)
 - Double quotation marks ("")
 - Parentheses ()
 - and

v. not

vi. or

Operator	Description
and	The and operator. Example: <code>request_method:GET and status:200</code> . If no syntax keyword exists among multiple keywords, the keywords are joined by using the and operator by default. For example, <code>GET 200 cn-shanghai</code> is equivalent to <code>GET and 200 and cn-shanghai</code> .
or	The or operator. Example: <code>request_method:GET or status:200</code> .
not	The not operator. Examples: <code>request_method:GET not status:200</code> and <code>not status:200</code> .
()	This operator is used to increase the priority of the search conditions that are enclosed in parentheses (). Example: <code>(request_method:GET or request_method:POST) and status:200</code> .
:	This operator is used for field-specific searches based on the key:value format. Example: <code>request_method:GET</code> . 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: <code>"file info":apsara</code> .
""	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, <code>"and"</code> 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.
\	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 <code>instance_id:nginx"01"</code> , you can execute the <code>instance_id:nginx\"01\"</code> statement to search for the log.
*	The wildcard character. This character is used to match zero, one, or multiple characters. Example: <code>host:www.yl.mo*k.com</code> .  Note 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: <code>host:www.yl.mo?k.com</code> .
>	This operator is used to query the logs in which the value of a specified field is greater than a specified numeric value. Example: <code>request_time>100</code> .
>=	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: <code>request_time>=100</code> .
<	This operator is used to query the logs in which the value of a specified field is smaller than a specified numeric value. Example: <code>request_time<100</code> .

Operator	Description
<=	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: <code>request_time<=100</code> .
=	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, <code>request_time=100</code> is equivalent to <code>request_time:100</code> .
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: <code>request_time in [100 200]</code> and <code>request_time in (100 200]</code> .  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: <code>__source__:192.0.2.*</code> .  Notice The __source__ field is a reserved field in Log Service. This field can be abbreviated to source. If you configure a custom field in the source format, the custom field conflicts with the reserved source field in Log Service. If you want to search for the custom field, you must use Source or SOURCE in a search statement.
__tag__	This operator is used to query logs based on metadata. Example: <code>__tag__:__receive_time__:1609837139</code> .
__topic__	This operator is used to query the logs of a specified log topic. Example: <code>__topic__:nginx_access_log</code> .

Examples of search statements

Expected search result	Search statement
Logs that contain successful GET requests (status codes: 200 to 299)	<code>request_method:GET and status in [200 299]</code>
Logs that contain GET requests that are not sent from the China (Shanghai) region	<code>request_method:GET not region:cn-shanghai</code>
Logs that contain GET requests or POST requests	<code>request_method:GET or request_method:POST</code>
Logs that do not contain GET requests	<code>not request_method:GET</code>
Logs that contain successful GET requests or successful POST requests	<code>(request_method:GET or request_method:POST) and status in [200 299]</code>

Expected search result	Search statement
Logs that contain failed GET requests or failed POST requests	<code>(request_method:GET or request_method:POST) not status in [200 299]</code>
Logs that contain successful GET requests (status codes: 200 to 299) and in which the request duration is less than 60 seconds	<code>request_method:GET and status in [200 299] not request_time>=60</code>
Logs in which the request duration is equal to 60 seconds	<code>request_time:60</code> <code>request_time=60</code>
Logs in which the request duration is greater than or equal to 60 seconds and is less than 200 seconds	<code>request_time>=60 and request_time<200</code> <code>request_time in [60 200)</code>
Logs in which the value of the http_user_agent field contains Firefox	<code>http_user_agent:Firefox</code>
Logs in which the value of the http_user_agent field contains Linux and Chrome	<code>http_user_agent:"Linux Chrome"</code> <code>http_user_agent:Linux and http_user_agent:Chrome</code>
Logs that contain and	<code>"and"</code> 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	<code>http_user_agent:Firefox or http_user_agent:Chrome</code>
Logs in which the value of the file info field contains apsara	<code>"file info":apsara</code>
Logs that contain strings that start with cn	<code>cn*</code>
Logs in which the value of the region field starts with cn	<code>region:cn*</code>
Logs in which the value of the region field contains cn*	<code>region:"cn*"</code>

Expected search result	Search statement
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	<code>mo?la</code>
Logs that contain strings that start with mo, end with la, and contain zero, one, or more characters between mo and la	<code>mo*la</code>
Logs that contain strings that start with Moz and strings that start with Sa	<code>Moz* and Sa*</code>
Logs whose topic is HTTPS or HTTP	<code>__topic__:HTTPS or __topic__:HTTP</code>
Logs that are collected from the 192.0.2.1 host	<code>__tag__:__client_ip__:192.0.2.1</code> <code>__tag__:__client_ip__</code> indicates the IP address of the host from which logs are collected.
Logs in which the remote_user field is not empty	<code>not remote_user:""</code>
Logs in which the remote_user field is empty	<code>remote_user:""</code>
Logs that do not contain the remote_user field	<code>not remote_user:*</code>
Logs in which the value of the remote_user field is null	<code>not request_uri:"null"</code>
Logs that contain the remote_user field	<code>remote_user:*</code>
Logs in which the value of the request_uri field is /request/path-2	<code>request_uri:/request/path-2</code>
Logs in which the value of the city field is not Shanghai	<code>not city:Shanghai</code>
Logs in which the value of the path field starts with /learn but does not contain /learn/level	<code>path:/learn* not path:/learn/level</code>

5.4.7.2. 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.

Context

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 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  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.

Operation	Description
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.

5.4.7.3. 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.

Context

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 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**.
 - 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**.

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.

Note 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.

- 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 sub-pattern 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

- On the **LogReduce** tab, click **Log Compare**.
- 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	7,890	+7,687 3787%	kind: Event apiVersion: audit.k8s.io/v1beta1 metadata: {"creationTimestamp": "2019-04-11T02:10:15Z"} requestURI: /api/v1/namespaces/*/*/* verb: get user: {"username": "system:anonymous"} responseStatus: {"code": 200, "message": "GET /api/v1/namespaces/*/*/*"} requestReceivedTimestamp: 2019-04-11T02:10:15Z annotations: {"authorization.k8s.io/decision": "allow", "authorization.k8s.io/reason": ""}	5Minutes 4Hours 30Days	15Minutes 1Hour 1Week	Start Time: 2019-04-10 10:17:15 End Time: 2019-04-10 10:32:15 OK
2	2,841	2,955	+114 4.01%	kind: Event apiVersion: audit.k8s.io/v1beta1 metadata: {"creationTimestamp": "2019-04-11T02:10:15Z"} requestURI: /api/v1/namespaces/*/*/* verb: get user: {"username": "system:anonymous"} responseStatus: {"code": 200, "message": "GET /api/v1/namespaces/*/*/*"} requestReceivedTimestamp: 2019-04-11T02:10:15Z annotations: {"authorization.k8s.io/decision": "allow", "authorization.k8s.io/reason": ""}			
3	0	2,282	+2,282 New	kind: Event apiVersion: audit.k8s.io/v1beta1 metadata: {"creationTimestamp": "2019-04-11T02:10:15Z"} requestURI: /api/v1/namespaces/*/*/* verb: get user: {"username": "system:anonymous"} responseStatus: {"code": 200, "message": "GET /api/v1/namespaces/*/*/*"} requestReceivedTimestamp: 2019-04-11T02:10:15Z annotations: {"authorization.k8s.io/decision": "allow", "authorization.k8s.io/reason": ""}			
4	0	1,801	+1,801 New	kind: Event apiVersion: audit.k8s.io/v1beta1 metadata: {"creationTimestamp": "2019-04-11T02:10:15Z"} requestURI: /api/v1/namespaces/*/*/* verb: get user: {"username": "system:anonymous"} responseStatus: {"code": 200, "message": "GET /api/v1/namespaces/*/*/*"} requestReceivedTimestamp: 2019-04-11T02:10:15Z annotations: {"authorization.k8s.io/decision": "allow", "authorization.k8s.io/reason": ""}			
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 .						

Parameter	Description
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.diff 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**.

5.4.7.4. 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](#).

 **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 [Enable the indexing feature and configure indexes for a Logstore](#).

Context

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 log collected from the log file of the server. This helps you identify and resolve errors.

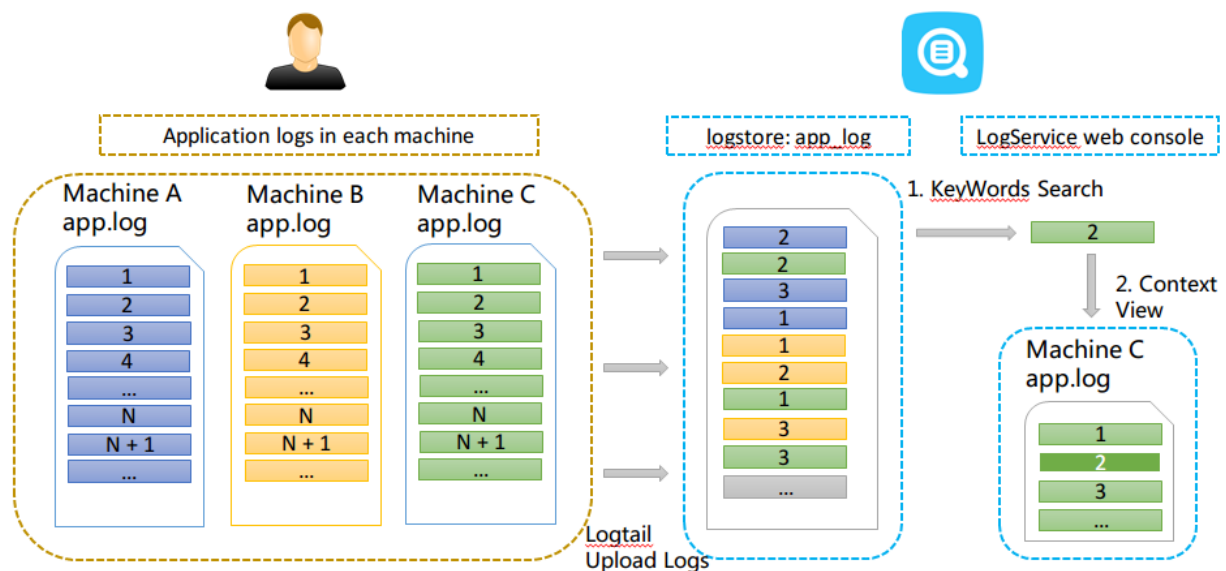
Scenario

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.

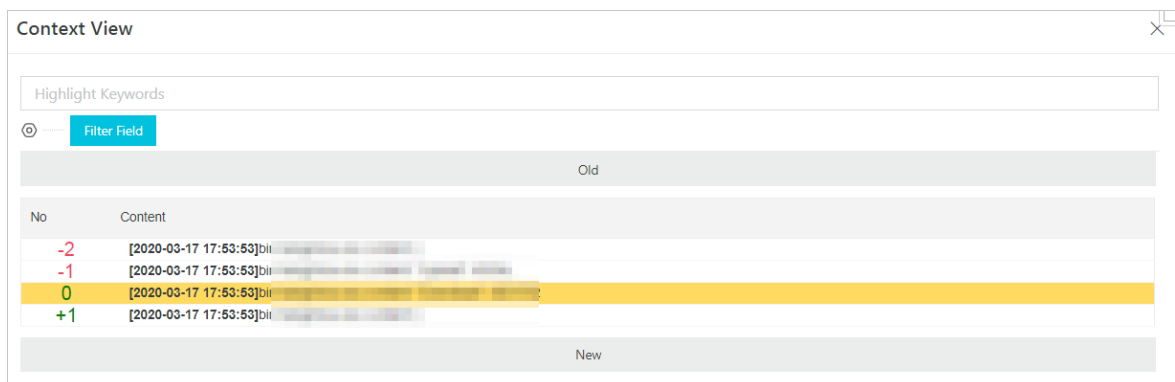
Benefits

- 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 Log Service console](#).
2. In the Projects section, click the project in which you want to perform contextual queries.
3. Click the  icon 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.



5.4.7.5. 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 [Enable the indexing feature and configure indexes for a Logstore](#).

Context

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 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. 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 Details

* Saved Search Name

stage

Attributes

Logstores

audit-c03ff60740131455e931115eb83832ea8

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 placeholder variable. You can configure a drill-down configuration to replace the variable.

Variable Config

Variable Name:

stage

Default Value:

stage

Matching Mode:

Global Match

Result

* | SELECT **S(stage)**, COUNT(*) as number GROUP BY **\$stage** LIMIT 10

Parameter	Description
Saved Search Name	The name of the saved search. The name must be 3 to 63 characters in length.
Variable Config	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. 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 Drill-down analysis. ? 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  icon 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.

5.4.7.6. 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 [Enable the indexing feature and configure indexes for a Logstore](#).

Features

- Allows you to analyze the first 100,000 log entries that are returned for a query.

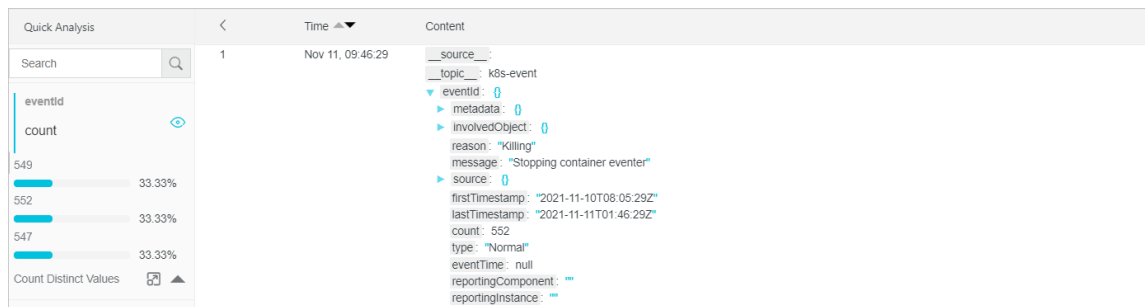
 **Note** When you perform quick analysis on log entries within a specified time range, Log Service collects the first 100,000 log entries. If you use a saved search to query all data in a Logstore, you must delete the Limit 100000 clause.

- 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.
5. Click the  icon to perform quick analysis based on the current query time range and view the distribution of

the field in logs.



Text type

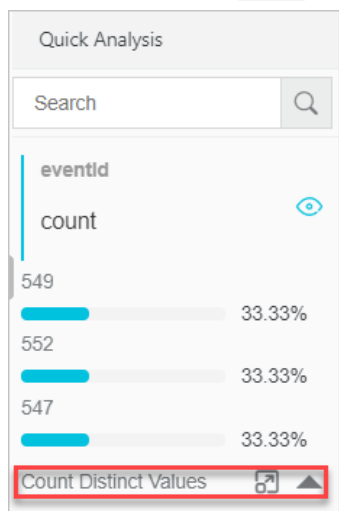
- Group and analyze log data by field of the text type.

Click the Eye icon next to a field of the **text** type to group the first 100,000 log entries and obtain the percentages of the top 10 groups.

The following example shows a query statement:

```
$Search | select ${keyName} , pv, pv *1.0/sum(pv) over() as percentage from( select count(1) as pv , "${keyName}" from (select "${keyName}" from log limit 100000) group by "${keyName}" order by pv desc) order by pv desc limit 10
```

The following figure shows the result that is returned after log entries are grouped by the `request_method` field. In this example, `GET` requests account for the majority of all returned requests.



- Calculate the number of unique values of a field.

Under the specified field in the **Quick Analysis** column, click **Count Distinct Values** to calculate the number of unique values of the `${keyName}` field.

- Automatically copy the query statement that is used to group and analyze data to the Search & Analyze search box.

Click the  icon next to **Count Distinct Values**. The query statement that is used to group and analyze data is automatically copied to the Search & Analyze search box. You can modify the query statement.

Long and double types

- Display approximate distribution by using histograms.

A large number of values exist for fields of the `long` and `double` types. Therefore, the preceding grouping and analysis method is not suitable for these fields. You can use the following query statement to divide field values into 10 buckets and display the approximate distribution of the values in a histogram:

```
$Search | select numeric_histogram(10, ${keyName})
```

The following figure shows the approximate distribution of the values in the `request_time` field. In this example, the largest percentage of request time is distributed at approximately 0.059 seconds.

- Perform quick analysis by using the `Max`, `Min`, `Avg`, and `Sum` clauses.

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 the values.

- Automatically copy the query statement that is used to calculate the approximate distribution to the Search & Analyze search box.

Click the  icon next to `Sum`. The query statement that is used to calculate the approximate distribution is automatically copied to the Search & Analyze search box. You can modify the query statement.

5.4.7.7. Other features

Log Service allows you to query log data by using various statements. It also provides many features for data search and analysis. This topic describes the raw log query, chart, saved search, dashboard, quick analysis, and alert features that Log Service supports.

Raw log query

After the index feature is enabled, enter one or more keywords in the search box and set the query time range. Then, click **Search & Analyze** to view the number of log entries displayed in a histogram, the raw logs, and the available charts.

The histogram shows the time-based distribution of log entry occurrences. In the histogram, you can view the changes in the number of matched log entries over a period of time. You can click a rectangular to view the information about the occurrences within the specified time.

On the Raw Logs tab, you can view the matched log data in chronological order.

- You can click the triangle symbol next to **Time** to switch between the **chronological order** and **reverse chronological order**.
- You can click **Display Content Column** and then you can select **New Line** or **Full Line** to display the log entries. You can also set **Truncate Character String**.
- You can click the **keyword** of a field value in the log content to view all log entries that contain this keyword.
- You can click the  icon in the upper-right corner of the Raw Logs tab to download the query results in the CSV format. You can also click **Column Settings** to select fields from the displayed raw log entries. This provides you with an easier way to view target field values of each raw log entry in new columns.
- You can click **Context View** to view the 15 log entries before and after the current log entry in a log file. For more information, see [Contextual query](#).

 **Note** The contextual query feature applies only to the data that is uploaded by using Logtail.



Charts

After you have enabled the index feature and entered a search and analytic statement, you can select a chart on the **Graph** tab to visualize the query results.

- Multiple types of charts are provided, such as the table and line chart.
You can select a chart type and configure the chart based on your business requirements.
- You can add charts to a **dashboard**. For more information, see [Create and delete a dashboard](#).
- You can configure drill-down analysis for a chart. After you add a chart to a dashboard, you can click the chart data to trigger a drill down event and explore in-depth details about the data. For more information about how to configure drill-down analysis, see [Drill-down analysis](#).

Dashboards

Log Service provides the dashboard feature that can visualize the results of search and analytic statements. For more information, see [Create and delete a dashboard](#).

Alerts

Log Service allows you to configure alerts for your dashboard or the charts on your dashboard. You can configure rules to receive alerts from Log Service.

For more information, see [Configure an alert rule](#).

5.4.8. SQL syntax and functions

5.4.8.1. 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: <pre>* SELECT arbitrary(request_method) AS request_method</pre>

Function	Description	Example
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. <pre>method: PostLogstoreLogs SELECT avg(latency) AS avg_latency, Project GROUP BY Project HAVING avg_latency > 1000</pre>
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: <pre>* SELECT checksum(request_method)</pre> The returned result is <code>D2UmTL3octI=</code>.
count(*)	Calculates the number of log entries.	The following query statement calculates the number of page views (PVs): <pre>* SELECT count(*) AS PV</pre>
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: <pre>* SELECT count(request_method)</pre>
count(1)	Calculates the number of log entries. This function is equivalent to <code>count(*)</code> .	The following query statement calculates the number of PVs: <pre>* SELECT count(1) AS PV</pre>
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. <pre>* SELECT count_if(url like '%abc')</pre>
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: <pre>* SELECT geometric_mean(request_time)</pre>

Function	Description	Example
<code>max_by(KEY_01,KEY_02)</code>	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: <pre>* SELECT max_by(UsageEndTime, PretaxAmount) as time</pre>
<code>max_by(KEY_01,KEY_02,n)</code>	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: <pre>* SELECT max_by(request_method,request_time,3)</pre> The returned result is <pre>["GET", "PUT", "DELETE"]</pre>
<code>min_by(KEY_01,KEY_02)</code>	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: <pre>* SELECT min_by(request_method,request_time)</pre>
<code>min_by(KEY_01,KEY_02,n)</code>	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: <pre>* SELECT min_by(request_method,request_time,3)</pre> The returned result is <pre>["GET", "PUT", "DELETE"]</pre>
<code>max(KEY)</code>	Queries the maximum value of a specified column.	The following query statement queries the longest request duration: <pre>* SELECT max(request_time)</pre>
<code>min(KEY)</code>	Queries the minimum value of a specified column.	The following query statement queries the shortest request duration: <pre>* SELECT min(request_time)</pre>
<code>sum(KEY)</code>	Calculates the total value of a specified column.	The following statement calculates the total size of daily NGINX traffic: <pre>* select date_trunc('day',__time__) AS time, sum(body_bytes_sent) AS body_bytes_sent GROUP BY time ORDER BY time</pre>

Function	Description	Example
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)

5.4.8.2. 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>

Function	Description	Example
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	country	provider	PV
180	CN	E	3
103	CN		3
180	CN	E	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

alarm5/64

* Add to New Dashboard

Create

access_alarm12/64

* Chart Name

alarm5/64

Query

* | 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

* Search Period

15 Minutes(Relative)

* Check Frequency

Fixed Interval

15

+

-

Minutes

* Trigger Condition

pv>54/128

Five basic operators are supported: plus (+), minus (-), multiplication (*), division (/), and modulo (%). 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 negated regex match (!~).[Documentation](#)

Advanced >

Next

Cancel

5.4.8.3. 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.

Notice 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(x)	Returns the size of a map.
element_at function	element_at(x, key)	Returns the value of a key from a map.

Function	Syntax	Description
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.
map function	map()	Returns an empty map.
	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 <i>x</i> and <i>y</i> . <i>x</i> is a key in the map. <i>y</i> is the value of the key in the map. If <i>y</i> has multiple values, a random value is extracted as the value of the key.
map_concat function	map_concat(<i>x</i> , <i>y</i> ...)	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(<i>x</i>)	Returns an array that consists of all values in a map.
multimap_agg function	multimap_agg(<i>x</i> , <i>y</i>)	Returns a multimap that is created by using <i>x</i> and <i>y</i> . <i>x</i> is a key in the multimap. <i>y</i> is the value of the key in the multimap. The value is of the array type. If <i>y</i> 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
<i>x</i>	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

```
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" }
```

- Query statement

```
* | SELECT try_cast(json_parse(etl_context) AS map(varchar, varchar))['project']
```

- Query and analysis result

_col0
datalab-148****6461-cn-chengdu

cardinality function

The cardinality function is used to return the size of a map.

- Syntax

```
cardinality(x)
```

- Parameters

Parameter	Description
<i>x</i>	The value of this parameter is of the map type.

- Return value type

The bigint type.

- Example

Use the histogram function to obtain the number of requests for each request method. Then, use the cardinality function to obtain the number of request methods.

- Query statement

```
* |
SELECT
  histogram(request_method) AS request_method,
  cardinality(histogram(request_method)) AS "kinds"
```

- Query and analysis result

request_method	kinds
{"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
<i>x</i>	The value of this parameter is of the map type.
<i>key</i>	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.

- Query statement

```
* |
SELECT
  histogram(request_method) AS request_method,
  element_at(histogram(request_method), 'DELETE') AS "count"
```

- Query and analysis result

request_method	count
{"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
<i>x</i>	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

```
* | SELECT histogram(request_method) AS request_method
```

- Query and analysis result

request_method
{"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
<i>x</i>	The value of this parameter is of an arbitrary data type.

- Return value type

The bigint type.

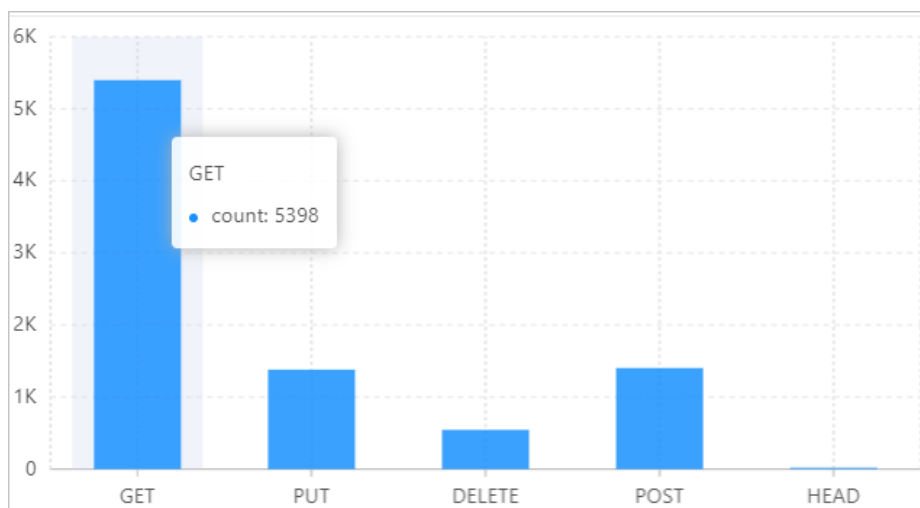
- Example

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

- 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
<i>x</i>	The value of this parameter is of the array type.

Parameter	Description
<i>y</i>	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	
	{"class01":49,"class03":45,"class02":50,"class05":50,"class04":47}

- Example 2: Return an empty map.

- Query statement

```
* | SELECT map()
```

- Query and analysis result

_col0	
	{}

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
<i>x</i>	The value of this parameter is of an arbitrary data type.
<i>y</i>	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

```
request_method:POST
request_time:80
```

- Query statement

```
* | 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

The `map_concat` function is used to return the union of multiple maps.

- Syntax

```
map_concat(x, y)
```

- Parameters

Parameter	Description
<i>x</i>	The value of this parameter is of the map type.
<i>y</i>	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.

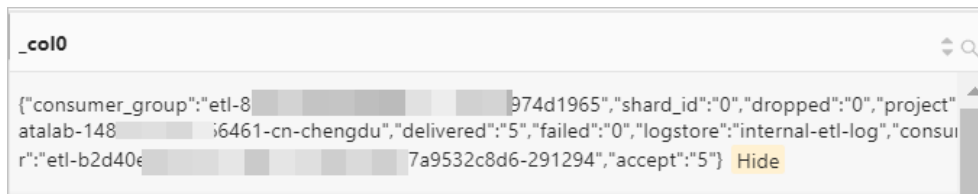
- Sample fields

```
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(vvarchar, varchar)
    ),
    cast (json_parse(progress) AS map(vvarchar, varchar))
  )
```

- Query and analysis result



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
<i>x</i>	The value of this parameter is of the map type.
<i>lambda_expression_expression</i>	The lambda expression. For more information, see Lambda functions .

- Return value type

The map type.

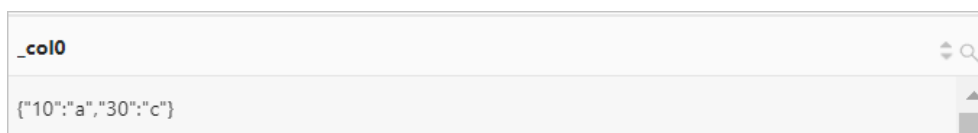
- Example

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)
```

- Query and analysis result



map_keys function

The `map_keys` function is used to return an array that consists of all keys in a map.

- Syntax

```
map_keys(x)
```

- Parameters

Parameter	Description
<i>x</i>	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"
  shard_id:"0" }
```

- Query statement

```
* | SELECT map_keys(try_cast(json_parse(etl_context) AS map(vvarchar, varchar)))
```

- Query and analysis result

_col0
["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
<i>x</i>	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_values` function to obtain the values of 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"
  shard_id:"0" }
```

- Query statement

```
* | SELECT map_values(try_cast(json_parse(etl_context) AS map(varchar, varchar)))
```

- Query and analysis result

_col0
["etl-85d' f85840-834336","etl-11434 3e41 c","rds_log","datalab-148 66461-cn-chengdu","0"] Hide

multimap_agg function

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
<i>x</i>	The value of this parameter is of an arbitrary data type.
<i>y</i>	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
["HEAD":["35.0,47.0,49.0,19.0,16.0,50.0,40.0,65.0,63.0,13.0,43.0,44.0,14.0,11.0],DELETE":["27.0,31.0,27.0,63.0,42.0,73.0,36.0,48.0,62.0,15.0,65.0,15.0,47.0,69.0,74.0,43.0,26.0,66.0,69.0,70.0,47.0,23.0,11.0,48.0,29.0,33.0,21.0,13.0,78.0,49.0,46.0,37.0,11.0,10.0,36.0,79.0,25.0,65.0,12.0,54.0,15.0,76.0,68.0,16.0,15.0,78.0,58.0,70.0,60.0,42.0,12.0,31.0,38.0,14.0,60.0,23.0,21.0,37.0,23.0,50.0,47.0,13.0,30.0,40.0,36.0,44.0,79.0,58.0,33.0,14.0,63.0,25.0,54.0,63.0,46.0,22.0,66.0,55.0,19.0,72.0,49.0,75.0,63.0,25.0,30.0,14.0,21.0,74.0,40.0,43.0,13.0,39.0,74.0,61.0,50.0,26.0,14.0,18.0,37.0,80.0,19.0,57.0,49.0,70.0,37.0,63.0,50.0,10.0,60.0,40.0,20.0,67.0,29.0,60.0,27.0,28.0,61.0,11.0,72.0,59.0,59.0,54.0,33.0,37.0,59.0,31.0,78.0,79.0,37.0,35.0,23.0,54.0,39.0,62.0,44.0,15.0,50.0,16.0,11.0,74.0,75.0,21.0,61.0,37.0,27.0,29.0,53.0,11.0,60.0,73.0,22.0,68.0,21.0,60.0,61.0,31.0,78.0,23.0,68.0,49.0,74.0,21.0,23.0,67.0,31.0,17.0,44.0,63.0,28.0,78.0,37.0,40.0,12.0,18.0,77.0,22.0,41.0,12.0,59.0,56.0,22.0,62.0,72.0,45.0,31.0,74.0,40.0,11.0,42.0,12.0,63.0,78.0,79.0,26.0,37.0,28.0,16.0,12.0,69.0,65.0,34.0,... Show

5.4.8.4. Approximate functions

This topic describes the syntax of approximate functions. This topic also provides examples on how to use the functions.

Function	Description	Example
<code>approx_distinct(x)</code>	Estimates the number of unique values in the x field.	None
<code>approx_percentile(x,percentage)</code>	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.
<code>approx_percentile(x,percentages)</code>	This function is similar to <code>approx_percentile(x,percentage)</code> . You can specify multiple percentages to return the values at the specified percentage positions.	<code>approx_percentile(x,array[0.1,0.2])</code>
<code>numeric_histogram(buckets,Value)</code>	Distributes all values of the <i>Value</i> field to multiple buckets. The <i>buckets</i> 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 <code>select count group by</code> . Note The returned result is of the JSON type.	<pre>method:POST select numeric_histogram(10,latency) :</pre> distributes the values of the latency field for POST requests to 10 buckets and calculates the number of latency field values in each bucket.
<code>numeric_histogram_u(buckets,Value)</code>	Distributes all values of the <i>Value</i> field to multiple buckets. The <i>buckets</i> 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 <code>select count group by</code> . Note The returned result is a table that includes multiple rows and columns.	<pre>method:POST select numeric_histogram(10,latency) :</pre> distributes the values of the latency field for POST requests to 10 buckets and calculates the number of latency field values in each bucket.

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.

5.4.8.5. 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
<code>corr(key1, key2)</code>	Calculates the correlation coefficient between two specific columns. The return value is in the range of [0,1].
<code>covar_pop(key1, key2)</code>	Calculates the population covariance of two specific columns.
<code>covar_samp(key1, key2)</code>	Calculates the sample covariance of two specific columns.
<code>regr_intercept(key1, key2)</code>	Returns the linear regression intercept of input values. <i>key1</i> is the dependent value and <i>key2</i> is the independent value.
<code>regr_slope(key1, key2)</code>	Returns the linear regression slope of input values. <i>key1</i> is the dependent value and <i>key2</i> is the independent value.
<code>stddev(key)</code>	Calculates the sample standard deviation of the <i>key</i> column. This function is equivalent to the <code>stddev_samp</code> function.
<code>stddev_samp(key)</code>	Calculates the sample standard deviation of the <i>key</i> column.
<code>stddev_pop(key)</code>	Returns the population standard deviation of the <i>key</i> column.
<code>variance(key)</code>	Calculates the sample variance of the <i>key</i> column. This function is equivalent to the <code>var_samp</code> function.
<code>var_samp(key)</code>	Calculates the sample variance of the <i>key</i> column.
<code>var_pop(key)</code>	Calculates the population variance of the <i>key</i> column.

Examples

- Example 1: Calculate the correlation coefficient of two specific columns.

- Query statement

```
* | SELECT corr(request_length, request_time)
```

- Query and analysis result

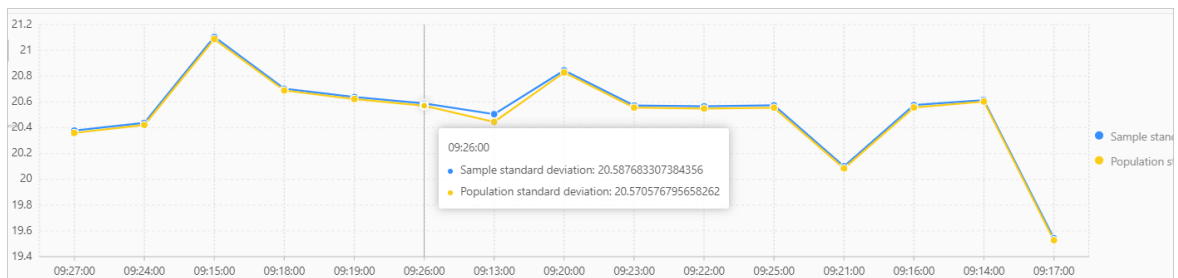
_col0
0.0008096234574114261

- 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", time_series(__time__, '1m', '%H:% I:%s', '0') AS time GROUP BY time
```


◦ Query and analysis result



5.4.8.6. 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: `+-*/%`.
- In the following functions, *x* and *y* can be numbers, log fields, or expressions whose calculation result is a number.

Function	Description
<code>abs(x)</code>	Calculates the absolute value of a number.
<code>cbrt(x)</code>	Calculates the cube root of a number.
<code>sqrt(x)</code>	Calculates the square root of a number.
<code>cosine_similarity(x,y)</code>	Calculates the cosine similarity between <i>x</i> and <i>y</i> .
<code>degrees(x)</code>	Converts radians to degrees.
<code>radians(x)</code>	Converts degrees to radians.
<code>e()</code>	Returns Euler's number.
<code>exp(x)</code>	Returns Euler's number raised to the power of a number.
<code>ln(x)</code>	Calculates the natural logarithm of a number.
<code>log2(x)</code>	Calculates the base-2 logarithm of a number.
<code>log10(x)</code>	Calculates the base-10 logarithm of a number.
<code>log(x,b)</code>	Calculates the base-b logarithm of a number.
<code>pi()</code>	Returns the value of π to 14 decimal places.
<code>pow(x,b)</code>	Calculates the value of a number raised to the power of <i>b</i> .
<code>rand()</code>	Returns a random number.
<code>random(0,n)</code>	Returns a random number that is greater than or equal to 0 and less than <i>n</i> .

Function	Description
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 <code>* SELECT floor(2.5)</code> statement, 2.0 is returned.
ceiling(x)	Returns a number rounded up to the nearest integer. For example, when you execute the <code>* SELECT ceiling(2.5)</code> statement, 3.0 is returned.
from_base(vvarchar, 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

5.4.8.7. 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
<code>chr(<i>number</i>)</code>	Returns the characters that match the ASCII value of a specified parameter.
<code>codepoint(<i>key</i>)</code>	Converts a field of the ASCII type to a field value of the bigint type.
<code>length(<i>key</i>)</code>	Calculates the length of a string. The return value is of the integer type.
<code>lower(<i>key</i>)</code>	Converts the characters in a string to lowercase letters. The return value is of the varchar type in lowercase letters.
<code>upper(<i>key</i>)</code>	Converts the characters in a string to lowercase letters. The return value is of the varchar type in uppercase letters.
<code>lpad(<i>key</i>, <i>length</i>, <i>lpad_string</i>)</code>	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.
<code>rpadd(<i>key</i>, <i>length</i>, <i>rpadd_string</i>)</code>	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.
<code>trim(<i>key</i>)</code>	Deletes space characters from the start and the end of a string. The return value is of the varchar type.

Function	Description
<code>ltrim(<i>key</i>)</code>	Deletes space characters from the start of a string. The return value is of the varchar type.
<code>rtrim(<i>key</i>)</code>	Deletes space characters from the end of a string. The return value is of the varchar type.
<code>replace(<i>key</i>,<i>substring</i>,<i>replace</i>)</code>	Replaces matched characters in a string with specified characters. The return value is of the varchar type.
<code>replace(<i>key</i>,<i>substring</i>)</code>	Deletes matched characters from a string. The return value is of the varchar type.
<code>reverse(<i>key</i>)</code>	Reverses the characters in a string.
<code>split(<i>key</i>,<i>delimiter</i>,<i>N</i>)</code>	Splits a string with a specified delimiter and returns a set of <i>N</i> substrings. The return value is an array.
<code>split_part(<i>key</i>,<i>delimiter</i>,<i>part</i>)</code>	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.
<code>split_to_map(<i>key</i>, <i>delimiter01</i>, <i>delimiter02</i>)</code>	Splits a string with the first specified delimiter, and then splits the string with the second specified delimiter. The return value is a map.
<code>position(<i>substring</i> IN <i>key</i>)</code>	Returns the position of a specified substring in a string. The return value is of the integer type. The value starts from 1.
<code>strpos(<i>key</i>, <i>substring</i>)</code>	Returns the position of a specified substring in a string. This function is equivalent to the <code>position(<i>substring</i> IN <i>key</i>)</code> function. The return value is of the integer type. The value starts from 1.
<code>substr(<i>key</i>, <i>start</i>)</code>	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.
<code>substr(<i>key</i>, <i>start</i>, <i>length</i>)</code>	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.

Function	Description
<code>concat(key01,key02,key03)</code>	Concatenates multiple strings into one string. The return value is of the integer type. The value starts from 1.
<code>levenshtein_distance(key01, key02)</code>	Returns the minimum edit distance between two strings.
<code>hamming_distance(string1,string2)</code>	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/4.0dp1 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	Path
4355	/request/path-1/file-5
4328	/request/path-1/file-1
4296	/request/path-2/file-3

- 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.

```
* | SELECT substr(server_protocol,1,4) AS protocol, count(*) AS count GROUP BY server_protocol
```

protocol	count
HTTP	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
{"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
true
true

- Return the position of the letter H in the value of the server_protocol field.

```
* | SELECT strpos(server_protocol, 'H')
```

_col0
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.

```
* | SELECT split(server_protocol, '/', 2)
```

_col0
["HTTP", "2.0"]

- Replace cn in the region field with China.

```
* | select replace(region, 'cn', 'China')
```

_col0
China-shanghai
China-shanghai

5.4.8.8. 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: <code>YYYY-MM-DD</code> . Example: 2021-01-12. - Return value type: date.	<pre>* select current_date</pre>
current_time	Returns the current time. - Return value format: <code>HH:MM:SS.Ms Time zone</code> . Example: 01:14:51.967 Asia/Shanghai. - Return value type: time.	<pre>* select current_time</pre>
current_timestamp	Returns the current date and time. - Return value format: <code>YYYY-MM-DD HH:MM:SS.Ms Time zone</code> . Example: 2021-01-12 17:16:09.035 Asia/Shanghai. - Return value type: timestamp.	<pre>* select current_timestamp</pre>
current_timezone()	Returns the current time zone. Return value type: varchar. Example: Asia/Shanghai.	<pre>* select current_timezone()</pre>
localtime	Returns the local time. - Return value format: <code>HH:MM:SS.Ms</code> - Return value type: time.	<pre>* select localtime</pre>
localtimestamp	Returns the local date and time. - Return value format: <code>YYYY-MM-DD HH:MM:SS.Ms</code> - Return value type: timestamp.	<pre>* select localtime</pre>
now()	Returns the current date and time. This function is equivalent to the current_timestamp function. - Return value format: <code>YYYY-MM-DD HH:MM:SS.Ms Time zone</code> . - Return value type: timestamp.	<pre>* select now()</pre>
from_iso8601_timestamp(<i>ISO8601</i>)	Converts an ISO 8601-formatted datetime expression to a timestamp expression that contains a time zone. - Return value format: <code>YYYY-MM-DD HH:MM:SS.Ms Time zone</code> . - 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: <code>YYYY-MM-DD</code> - Return value type: date.	<pre>* select from_iso8601_date('2020-05-03')</pre>

Function	Description	Example
<code>from_unixtime(<i>UNIX timestamp</i>)</code>	Converts a UNIX timestamp to a timestamp expression. - Return value format: <code>YYYY-MM-DD HH:MM:S</code> <code>S.Ms</code> - Return value type: timestamp.	<pre>* select from_unixtime(1494985275)</pre>
<code>from_unixtime(<i>UNIX timestamp, time zone</i>)</code>	Converts a UNIX timestamp to a timestamp expression that contains a time zone. - Return value format: <code>YYYY-MM-DD HH:MM:S</code> <code>S.Ms Time zone</code> - Return value type: timestamp.	<pre>* select from_unixtime (1494985275, 'Asia/Shangh ai')</pre>
<code>to_unixtime(<i>timestamp</i>)</code>	Converts a timestamp expression to a UNIX timestamp. Return value type: long. Example: 1494985500.848.	<pre>* select to_unixtime('2017-05-17 09:45:00.848 Asia/Shanghai')</pre>

Time functions

Function	Description	Example
<code>date_format(<i>timestamp, format</i>)</code>	Converts a timestamp expression to a datetime format string.	<pre>* select date_format (date_parse('2017-05-17 09:45:00', '%Y-%m-%d %H:%i:%S'), '%Y-%m-%d')</pre>
<code>date_parse(<i>string, format</i>)</code>	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>

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.
%c	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.

format	Description
%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.
%l	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 <code>hh:mm:ss AM/PM</code> format.
%S	The second. Valid values: 00 to 59.
%s	The second. Valid values: 00 to 59.
%T	The time. The 24-hour clock is used. The time is in the <code>hh:mm:ss</code> 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.
%y	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
<code>* select date_trunc('second', 2021-01-12 03:04:05.000)</code>	<code>2021-01-12 03:04:05.000</code>	None.
<code>* select date_trunc('minute', 2021-01-12 03:04:05.000)</code>	<code>2021-01-12 03:04:00.000</code>	None.
<code>* select date_trunc('hour', 2021-01-12 03:04:05.000)</code>	<code>2021-01-12 03:00:00.000</code>	None.
<code>* select date_trunc('day', 2021-01-12 03:04:05.000)</code>	<code>2021-01-12 00:00:00.000</code>	Returns <code>00:00:00.000</code> of the specified date.
<code>* select date_trunc('week', 2021-01-12 03:04:05.000)</code>	<code>2021-01-11 00:00:00.000</code>	Returns <code>00:00:00.000</code> of the Monday of the specified week.
<code>* select date_trunc('month', 2021-01-12 03:04:05.000)</code>	<code>2021-01-01 00:00:00.000</code>	Returns <code>00:00:00.000</code> of the first day of the specified month.
<code>* select date_trunc('quarter', 2021-01-11 03:04:05.000)</code>	<code>2021-01-01 00:00:00.000</code>	Returns <code>00:00:00.000</code> of the first day of the specified quarter.
<code>* select date_trunc('year', 2021-01-11 03:04:05.000)</code>	<code>2021-01-01 00:00:00.000</code>	Returns <code>00:00:00.000</code> 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

Function	Description	Example
<code>date_add(unit, N,timestamp)</code>	Adds N units to a <code>timestamp</code> . 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')</pre> Indicates seven days before August 9, 2018 (2018-08-02 00:00:00.000).
<code>date_diff(unit, timestamp1, timestamp2)</code>	Returns the time difference between two time expressions. For example, you can calculate the difference between <code>timestamp1</code> and <code>timestamp2</code> 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
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.

 **Notice** 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
<code>time_column</code>	The sequence of time (KEY), for example, <code>__time__</code>. The value of this parameter can be a long datetime or timestamp expression.

Parameter	Description
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
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

5.4.8.9. 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.

Function	Description	Example	
		Query statement	Query result
<code>url_extract_fragment(url)</code>	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/dashboard-demo/categoryList')</pre>	<code>/project/dashboard-demo/categoryList</code>

Function	Description	Example	
		Query statement	Query result
<code>url_extract_host(url)</code>	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>	<code>www.aliyun.com</code>
<code>url_extract_parameter(url, name)</code>	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>	<code>testuser</code>
<code>url_extract_path(url)</code>	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>	<code>/product/sls</code>
<code>url_extract_port(url)</code>	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>	<code>80</code>
<code>url_extract_protocol(url)</code>	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>	<code>http</code>
<code>url_extract_query(url)</code>	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>	<code>userid=testuser</code>
<code>url_encode(value)</code>	Encodes a URL.	<pre>* select url_encode('http://www.aliyun.com:80/product/sls?userid=testuser')</pre>	<code>http%3a%2f%2fwww.aliyun.com%3a80%2fproduct%2fsls%3fuserid%3dtestuser</code>
<code>url_decode(value)</code>	Decodes a URL.	<pre>* select url_decode('http%3a%2f%2fwww.aliyun.com%3a80%2fproduct%2fsls%3fuserid%3dtestuser')</pre>	<code>http://www.aliyun.com:80/product/sls?userid=testuser</code>

5.4.8.10. 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
<code>regexp_extract_all(string, pattern)</code>	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 <code>* SELECT regexp_extract_all('5a 67b 890m', '\d+')</code> is <code>['5', '67', '890']</code> . The returned result of <code>* SELECT regexp_extract_all('5a 67a 890m', '(\d+)a')</code> is <code>['5a', '67a']</code> .
<code>regexp_extract_all(string, pattern, group)</code>	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 <code>()</code> of a substring that derives from the specified string.	The returned result of <code>* SELECT regexp_extract_all('5a 67a 890m', '(\d+)a', 1)</code> is <code>['5', '67']</code> .
<code>regexp_extract(string, pattern)</code>	Returns the first substring of the specified string that matches the regular expression.	The returned result of <code>* SELECT regexp_extract('5a 67b 890m', '\d+')</code> is <code>'5'</code> .
<code>regexp_extract(string, pattern, group)</code>	Returns a part of the first substring that matches the regular expression. This part is the content in the group parameter value of the <code>()</code> of the substring that derives from the specified string.	The returned result of <code>* SELECT regexp_extract('5a 67b 890m', '(\d+) ([a-z]+)', 2)</code> is <code>'a'</code> .
<code>regexp_like(string, pattern)</code>	Returns a Boolean value. If the string and its substrings cannot match the regular expression, the value False is returned.	The returned result of <code>* SELECT regexp_like('5a 67b 890m', '\d+m')</code> is True.
<code>regexp_replace(string, pattern, replacement)</code>	Replaces the substrings of the specified string that match the regular expression with the value of the replacement parameter.	The returned result of <code>* SELECT regexp_replace('5a 67b 890m', '\d+', 'a')</code> is <code>'aa ab am'</code> .
<code>regexp_replace(string, pattern)</code>	Removes the substrings of the specified string that match the regular expression. This function is equivalent to <code>regexp_replace(string, pattern, '')</code> .	The returned result of <code>* SELECT regexp_replace('5a 67b 890m', '\d+')</code> is <code>'a b m'</code> .
<code>regexp_split(string, pattern)</code>	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 <code>* SELECT regexp_split('5a 67b 890m', '\d+')</code> is <code>['a', 'b', 'm']</code> .

5.4.8.11. 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 function](#).

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.

- Syntax

```
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.

- 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.

 **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

```
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) ) as 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.

5.4.8.12. 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

 **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)
```

5.4.8.13. 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.

 **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>

Function	Description	Example
ip_to_province(KEY)	Identifies the state to which an IP address belongs. The returned result is the Chinese name of a state.	<pre>* SELECT ip_to_province(client_ip)</pre>
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 <code>latitude,longitude</code> format. For information about geohash functions, see Geography functions .	<pre>* SELECT ip_to_geo(client_ip)</pre>
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 <code>latitude,longitude</code> 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.	<pre>* SELECT ip_to_provider(client_ip)</pre>

5.4.8.14. GROUP BY clause

GROUP BY clauses are used together with aggregate functions to group analysis results based on one or more columns.

Syntax

```
* | SELECT column name, aggregate function GROUP BY [ column name | alias | serial number ]
```

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. 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 Field aliases .
Serial number	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: <pre>* SELECT status, count(*) AS PV GROUP BY status</pre> <pre>* SELECT status, count(*) AS PV GROUP BY 1</pre>
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:

```
* | 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](#).

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 `__time__` 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
```

5.4.8.15. 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
<code>rank()</code>	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.
<code>row_number()</code>	Returns a unique, sequential number for each row.
<code>first_value(x)</code>	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.
<code>last_value(x)</code>	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.
<code>nth_value(x, offset)</code>	Returns the value at the specified offset from the beginning of the window.
<code>lead(x, offset, default_value)</code>	Returns the value in offset rows that follow the current row in the window. If the target row does not exist, the <code>default_value</code> is returned.

Function	Description
<code>lag(x,offset,default_value)</code>	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, personId, salary , rank() over(PARTITION BY department order by salary desc) as salary_rank order by department,salary_rank
```

Query and analysis result

department	personId	salary	salary_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, personId, salary *1.0 / sum(salary) over(PARTITION BY department ) as salary_percentage
```

Query and analysis result

department	personId	salary	salary_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

5.4.8.16. 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:

```
* | select avg(rain) ,province where temperature > 10 group by province having avg(rain) > 100
```

5.4.8.17. 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
```

5.4.8.18. 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
```

5.4.8.19. 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]
END

```

- 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	pv
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	pv
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
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
61 198
27 .181
11 .52
36 .48

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

5.4.8.20. Nested subquery

This topic describes how to use nested subqueries when you query logs.

You can use nested queries to perform more complicated queries.

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
)
```

5.4.8.21. 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.

 **Notice** 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	[<i>x</i>]	Returns the element whose index is <i>x</i> from an array. This operator is equivalent to the <code>element_at</code> function.
<code>array_agg</code> function	<code>array_agg(<i>x</i>)</code>	Returns an array that consists of all values in <i>x</i> .
<code>array_distinct</code> function	<code>array_distinct(<i>x</i>)</code>	Removes duplicate elements from an array.
<code>array_except</code> function	<code>array_except(<i>x</i>, <i>y</i>)</code>	Returns the difference between two arrays.
<code>array_intersect</code> function	<code>array_intersect(<i>x</i>, <i>y</i>)</code>	Returns the intersection of two arrays.
<code>array_join</code> function	<code>array_join(<i>x</i>, <i>delimiter</i>)</code>	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.  Notice The <code>array_join</code> 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.
	<code>array_join(<i>x</i>, <i>delimiter</i>, <i>null_replacement</i>)</code>	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 <code>null_replacement</code> parameter.  Notice The <code>array_join</code> 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.
<code>array_max</code> function	<code>array_max(<i>x</i>)</code>	Returns the maximum value in an array.
<code>array_min</code> function	<code>array_min(<i>x</i>)</code>	Returns the minimum value in an array.

Function	Syntax	Description
<code>array_position</code> function	<code>array_position(<i>x</i>, <i>element</i>)</code>	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.
<code>array_remove</code> function	<code>array_remove(<i>x</i>, <i>element</i>)</code>	Removes a specified element from an array.
<code>array_sort</code> function	<code>array_sort(<i>x</i>)</code>	Sorts the elements in an array in ascending order. If the array contains a null element, the null element is placed at the end.
<code>array_transpose</code> function	<code>array_transpose(<i>x</i>)</code>	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.
<code>array_union</code> function	<code>array_union(<i>x</i>, <i>y</i>)</code>	Returns the union of two arrays.
<code>cardinality</code> function	<code>cardinality(<i>x</i>)</code>	Returns the number of elements in an array.
<code>concat</code> function	<code>concat(<i>x</i>, <i>y</i>...)</code>	Concatenates multiple arrays into one array.
<code>contains</code> function	<code>contains(<i>x</i>, <i>element</i>)</code>	Checks whether an array contains a specified element. If the array contains the specified element, the function returns true.
<code>element_at</code> function	<code>element_at(<i>x</i>, <i>y</i>)</code>	Returns the element whose index is <i>y</i> from an array.
<code>filter</code> function	<code>filter(<i>x</i>, <i>lambda_expression</i>)</code>	Filters elements in an array based on a lambda expression and returns elements that match the lambda expression.
<code>flatten</code> function	<code>flatten(<i>x</i>)</code>	Transforms a two-dimensional array into a one-dimensional array.
<code>reduce</code> function	<code>reduce(<i>x</i>, <i>lambda_expression</i>)</code>	Returns the sum of the elements in an array based on a lambda expression.
<code>reverse</code> function	<code>reverse(<i>x</i>)</code>	Reverses the elements in an array.
<code>sequence</code> function	<code>sequence(<i>x</i>, <i>y</i>)</code>	Returns an array of elements within a specified range. The elements are consecutive and incremental. The default incremental step is 1.
	<code>sequence(<i>x</i>, <i>y</i>, <i>step</i>)</code>	Returns an array of elements within a specified range. The elements are consecutive and incremental. The incremental step is a custom value.
<code>shuffle</code> function	<code>shuffle(<i>x</i>)</code>	Shuffles the elements in an array.
<code>slice</code> function	<code>slice(<i>x</i>, <i>start</i>, <i>length</i>)</code>	Returns a subset of an array.

Function	Syntax	Description
transform function	<code>transform(<i>x</i>, <i>lambda_expression</i>)</code>	Transforms each element in an array by using a lambda expression.
zip function	<code>zip(<i>x</i>, <i>y</i>...)</code>	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	<code>zip_with(<i>x</i>, <i>y</i>, <i>lambda_expression</i>)</code>	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.

- Syntax

```
[x]
```

- Parameters

Parameter	Description
<i>x</i>	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 number field.

- Sample field

```
number: [49, 50, 45, 47, 50]
```

- Query statement

```
* | SELECT cast(json_parse(number) as array(bigint)) [1]
```

- Query and analysis result

_col0
49

array_agg function

The `array_agg` function is used to return an array that 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 status field.

- Query statement

```
* | SELECT array_agg(status) AS array
```

- Query and analysis result

[illegible]

array_distinct function

The `array_distinct` function is used to remove duplicate 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`

`[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
<i>x</i>	The value of this parameter is of the array type.
<i>y</i>	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])
```

- Query and analysis result

_col0
[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
<i>x</i>	The value of this parameter is of the array type.
<i>y</i>	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

```
* | SELECT array_intersect(array[1,2,3,4,5],array[1,3,5,7])
```

- Query and analysis result

_col0
[1,3,5]

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
<i>x</i>	The value of this parameter is of an arbitrary array type.
<i>delimiter</i>	The delimiter that is used to connect elements. You can specify a string for this parameter.
<i>null_replacement</i>	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

```
* | SELECT array_join(array[null, 'Log', 'Service'], ' ', 'Alicloud')
```

- Query and analysis result

_col0	
Alicloud Log Service	

array_max function

The `array_max` function is used to return the maximum value in an array.

- Syntax

```
array_max(x)
```

- Parameters

Parameter	Description
<i>x</i>	The value of this parameter is of the array type.  Notice If an array contains a null element, the function returns null.

- Return value type

The data type of elements in the parameter value.

- Example

Obtain the maximum value in an array.

- Sample field

```
number: [49, 50, 45, 47, 50]
```

- Query statement

```
*| SELECT array_max(try_cast(json_parse(number) as array(bigint))) AS max_number
```

- Query and analysis result

max_number
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.  Notice If an array contains a null element, the function returns null.

- Return value type

The data type of elements in the parameter value.

- Example

Obtain the minimum value in an array.

- Sample field

```
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
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
<i>x</i>	The value of this parameter is of the array type.
<i>element</i>	The value of this parameter is the element whose index you want to obtain. ? 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
2

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
<i>x</i>	The value of this parameter is of the array type.
<i>element</i>	The value of this parameter is the element that you want to remove. ? Note If the element is null, the function returns null.

- Return value type

The array type.

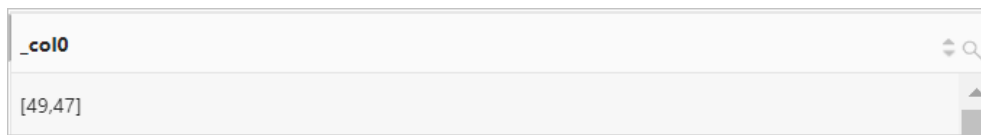
- Example

Remove 45 from the [49,45,47] array.

- Query statement

```
* | SELECT array_remove(array[49,45,47],45)
```

- Query and analysis result



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
<i>x</i>	The value of this parameter is of the array type.

- Return value type

The array type.

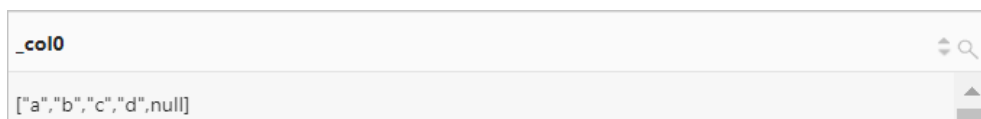
- Example

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



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)
```

- Parameters

Parameter	Description
<i>x</i>	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

```
* | SELECT array_transpose(array[array[0,1,2,3],array[10,19,18,17],array[9,8,7]])
```

- Query and analysis result

_col0
[[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
<i>x</i>	The value of this parameter is of the array type.
<i>y</i>	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.

- Query statement

```
* | SELECT array_union(array[1,2,3,4,5],array[1,3,5,7])
```

- Query and analysis result

_col0
[1,2,3,4,5,7]

cardinality function

The cardinality function is used to return the number of elements in an array.

- Syntax

```
cardinality(x)
```

- Parameters

Parameter	Description
<i>x</i>	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 number field.

- Sample field

```
number: [49,50,45,47,50]
```

- Query statement

```
*| SELECT cardinality(cast(json_parse(number) as array(bigint)))
```

- Query and analysis result

_col0
5

concat function

The concat function is used to concatenate multiple arrays into one array.

- Syntax

```
concat(x, y...)
```

- Parameters

Parameter	Description
<i>x</i>	The value of this parameter is of the array type.
<i>y</i>	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 concat(array['red','blue'],array['yellow','green'])
```

- Query and analysis result

_col0
["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
<i>x</i>	The value of this parameter is of the array type.

Parameter	Description
<i>element</i>	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
true

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
<i>x</i>	The value of this parameter is of the array type.
<i>y</i>	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]
```

- Query statement

```
* |
SELECT
  element_at(cast(json_parse(number) AS array(varchar)), 2)
```

- 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
<i>x</i>	The value of this parameter is of the array type.
<i>lambda_expression</i>	The lambda expression. For more information, see Lambda functions .

- 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 -> x > 0`.

- Query statement

```
* | SELECT filter(array[5,-6,null,7],x -> x > 0)
```

- Query and analysis result

_col0	⬇ 🔍
[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
<i>x</i>	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
<i>x</i>	The value of this parameter is of the array type.
<i>lambda_expression</i>	The lambda expression. For more information, see Lambda functions .

- 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) -> s + x, s -> s)
```

- Query and analysis result

_col0
75

reverse function

The reverse function is used to reverse the elements in an array.

- Syntax

```
reverse(x)
```

- Parameters

Parameter	Description
<i>x</i>	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
[5,4,3,2,1]

sequence function

The sequence function is used to return an array 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 function uses a custom incremental step:

```
sequence(x, y, step)
```

- Parameters

Parameter	Description
<i>x</i>	The value of this parameter is of the bigint or timestamp type. UNIX timestamps and date and time expressions are supported.
<i>y</i>	The value of this parameter is of the bigint or timestamp type. UNIX timestamps and date and time expressions are supported.
<i>step</i>	The incremental step. If the values of the <i>x</i> and <i>y</i> parameters are date and time expressions, the value of the <i>step</i> parameter is in one of the following formats: <code>interval 'n' year to month</code>: The incremental step is <i>n</i> years. <code>interval 'n' day to second</code>: The incremental step is <i>n</i> days.

- Return value type

The array type.

- Examples

- 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
[0,2,4,6,8,10]

- Example 2: Obtain the dates within the range from 2017-10-23 to 2021-08-12 at the incremental step of 1 year.

- Query statement

```
ww* | SELECT  sequence(from_unixtime(1508737026),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","2019-10-23 13:37:06.000","2020-10-23 13:37:06.000"]
```

- Example 3: Obtain the UNIX timestamps within the range from 1628733298 to 1628734085 at the incremental step of 60 seconds.

- Query statement

```
* | SELECT  sequence(1628733298,1628734085,60)
```

- Query and analysis result

```
_col0
[1628733298,1628733358,1628733418,1628733478,1628733538,1628733598,1628733658,1628733718,1628733778,1628733838,1628733898,1628733958,1628734018,1628734078] Hide
```

shuffle function

The shuffle function is used to shuffle the elements in an array.

- Syntax

```
shuffle(x)
```

- Parameters

Parameter	Description
<i>x</i>	The value of this parameter is of the array type.

- Return value type

The array type.

- Example

Shuffle the elements in the [1,2,3,4,5] array.

- Query statement

```
*| SELECT  shuffle(array[1,2,3,4,5])
```

- Query and analysis result

```
_col0
[3,1,2,4,5]
[5,1,2,4,3]
[2,5,3,1,4]
```

slice function

The slice function is used to return a subset of an array.

- Syntax

```
slice(x, start, length)
```

- Parameters

Parameter	Description
<i>x</i>	The value of this parameter is of the array type.
<i>start</i>	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.
<i>length</i>	The number of elements that you want to include in the subset.

- Return value type

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	
[4,5]	

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
<i>x</i>	The value of this parameter is of the array type.
<i>lambda_expression</i>	The lambda expression. For more information, see Lambda functions .

- 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)
```

- Query and analysis result

_col0
[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
<i>x</i>	The value of this parameter is of the array type.
<i>y</i>	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])
```

- Query and analysis result

_col0
[[1,"1b",1],[2,null,2],[3,"3b",3]]

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
<i>x</i>	The value of this parameter is of the array type.
<i>y</i>	The value of this parameter is of the array type.
<i>lambda_expression</i>	The lambda expression. For more information, see Lambda functions .

- Return value type

The array type.

- Example

Use the lambda expression `(x, y) -> x + y` to add the elements in the [1, 2] and [3, 4] arrays and return a new array.

- Query statement

```
SELECT zip_with(array[1,2], array[3,4], (x,y) -> x + y)
```

- Query and analysis result

<code>_col0</code>
<code>[4,6]</code>

5.4.8.22. 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 (<code> </code>)	The result of <code>a b</code> is <code>ab</code> .
<code>length(binary)</code>	Returns the length of a binary string in bytes. The return value is of the bigint type.
<code>concat(binary1, ..., binaryN)</code>	Concatenates binary strings. This function is equivalent to <code> </code> . The return value is of the varbinary type.
<code>to_base64(binary)</code>	Converts a binary string to a Base64 string. The return value is of the varchar type.
<code>from_base64(string)</code>	Converts a Base64 string to a binary string. The return value is of the varbinary type.
<code>to_base64url(binary)</code>	Converts a string to a URL-safe Base64 string. The return value is of the varchar type.
<code>from_base64url(string)</code>	Converts a URL-safe Base64 string to a binary string. The return value is of the varbinary type.
<code>to_hex(binary)</code>	Converts a binary string to a hexadecimal string. The return value is of the varchar type.
<code>from_hex(string)</code>	Converts a hexadecimal string to a binary string. The return value is of the varbinary type.
<code>to_big_endian_64(bigint)</code>	Converts a number to a binary string in big endian mode. The return value is of the varbinary type.
<code>from_big_endian_64(binary)</code>	Converts a binary string in big endian mode to a number. The return value is of the bigint type.

Function	Description
md5(binary)	Calculates the MD5 value of a binary string. The return value is of the varbinary type.
sha1(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.

5.4.8.23. 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.	<code>SELECT bit_count(9, 64)</code> returns 2. <code>SELECT bit_count(9, 8)</code> returns 2. <code>SELECT bit_count(-7, 64)</code> returns 62. <code>SELECT bit_count(-7, 8)</code> returns 6.
bitwise_and(x, y)	Returns the bitwise AND of x and 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 x and y in two's complement. The return value is of the bigint type.	None

5.4.8.24. 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)
```

 **Note** The `compare()` function can be used to compare the calculation results of multiple periods of time, for example, `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].

- Examples

- Calculate the ratio of the page views (PVs) of the current hour to the PVs of the same time period the day before.

Set the time range to **1 Hour(Time Frame)** and execute the following query statement. 86400 indicates the current time minus 86400 seconds (one day). log indicates the Logstore name.

```
* | SELECT compare(PV, 86400) FROM (SELECT count(*) AS PV FROM log)
```

The following figure shows the returned result.

_col0
[3337.0,3522.0,0.947473026689381]

- **3337.0** indicates the PVs of the current 1 hour, for example, Dec 25, 2020, 14:00:00 ~ Dec 25, 2020, 15:00:00.
- **3522.0** indicates the PVs of the same time period the day before, for example, Dec 24, 2020, 14:00:00 ~ Dec 24, 2020, 15:00:00.
- **0.947473026689381** indicates the ratio of the PVs of the current hour to the PVs of the same time period the day before.

To display the analysis result in multiple columns, you can 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))
```

The following figure shows the returned result.

today	yesterday	ratio
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.

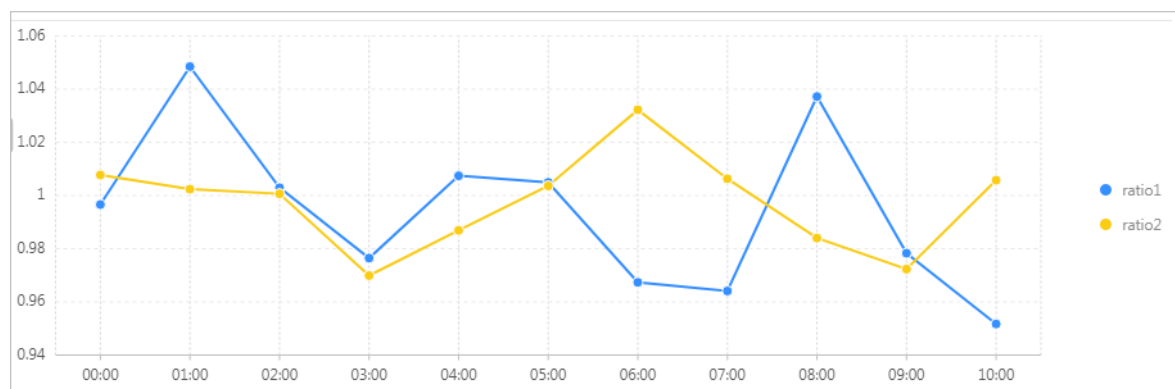
time	diff
00:00	[1176.0,1180.0,1167.0,0.9966101694915255,1.0077120822622108]
01:00	[10077.0,9611.0,10053.0,1.0484861096660077,1.0023873470605789]
02:00	[26921.0,26842.0,26903.0,1.002943148796662,1.0006690703638999]

- **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, 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.



- 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). log 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) GROUP BY time ORDER BY time
```

The following figure shows the returned result.

time	diff
2021-01-01 00:00:00.000	[11958378.0,448571.0,26.658829928818404]

ts_compare() function

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

```
ts_compare(column name,N)
```

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	data
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]

5.4.8.25. 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
```

5.4.8.26. 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 [Array functions](#), [Array functions](#), [Array functions](#), [Array functions](#), and [Map functions](#). Syntax:

```
parameter -> expression
```

Parameter	Description
<i>parameter</i>	The identifier that is used to pass parameters.
<i>expression</i>	The lambda expression, which can include most MySQL expressions. Examples: <pre> x -> x + 1 (x, y) -> x + y x -> regexp_like(x, 'a+') x -> x[1] / x[2] x -> if(x > 0, x, -x) x -> coalesce(x, 0) x -> cast(x AS JSON) x -> x + try(1 / 0) </pre>

Examples

- Example 1: $x \rightarrow x$ 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
[5,7]

- Example 2: $0, (s, x) \rightarrow s + x, s \rightarrow s$

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
75

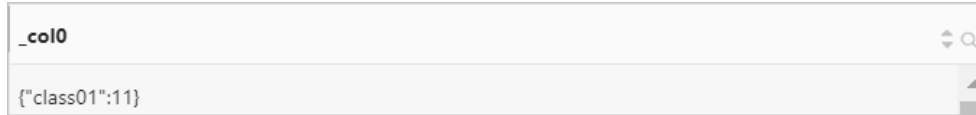
- Example 3: $(k,v) \rightarrow v > 10$

This lambda expression is used to create a map from two arrays. The values of keys in the map are greater than 10.

- Query statement

```
* | SELECT map_filter(map(array['class01', 'class02', 'class03'], array[11, 10, 9]), (k,v) -> v > 10)
```

- Query and analysis result



```
_col0
{"class01":11}
```

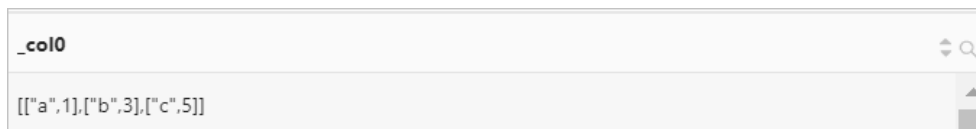
- Example 4: (x, y) -> (y, x)

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

```
* | SELECT zip_with(array[1, 3, 5], array['a', 'b', 'c'], (x, y) -> (y, x))
```

- Query and analysis result



```
_col0
[['a',1],['b',3],['c',5]]
```

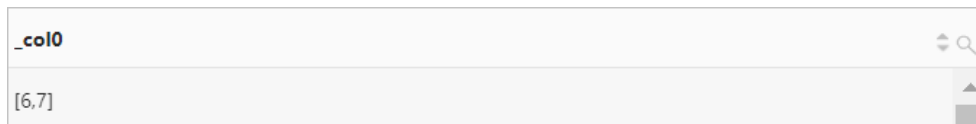
- Example 5: x -> coalesce(x, 0) + 1

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)
```

- Query and analysis result



```
_col0
[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)),
    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))
```

5.4.8.27. 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.

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

Truth table 2

a	NOT a
TRUE	FALSE
FALSE	TRUE
NULL	NULL

5.4.8.28. 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.

Sample aliases

Original column name	Alias
User-Agent	ua
User.Agent	ua
123	col
abceefghijklmnopqrstuvw	a

5.4.8.29. 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](#).

Context

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 syntax](#).

```
method:postlogstorelogs | select count(1) , histogram(logstore) from log l join join_meta m on l.
projectid = cast( m.ikey as varchar)
```

Notice

- 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 n
ame"));
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 me
ta 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();
```

5.4.8.30. 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.

Geometry formats

Geometry	WKT format
Point	POINT (0 0)
LineString	LINestring (0 0, 1 1, 1 2)

Geometry	WKT format
Polygon	<code>POLYGON ((0 0, 4 0, 4 4, 0 4, 0 0), (1 1, 2 1, 2 2, 1 2, 1 1))</code>
MultiPoint	<code>MULTIPOINT (0 0, 1 2)</code>
MultiLineString	<code>MULTILINESTRING ((0 0, 1 1, 1 2), (2 3, 3 2, 5 4))</code>
MultiPolygon	<code>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,</code>
GeometryCollection	<code>GEOMETRYCOLLECTION (POINT(2 3), LINESTRING (2 3, 3 4))</code>

Constructors

Constructor description

Function	Description
<code>ST_Point(double, double) → Point</code>	Returns a geometry point instance with the specified coordinate values.
<code>ST_LineFromText(varchar) → LineString</code>	Returns a geometry LineString instance from a WKT representation.
<code>ST_Polygon(varchar) → Polygon</code>	Returns a geometry polygon instance from a WKT representation.
<code>ST_GeometryFromText(varchar) → Geometry</code>	Returns a geometry instance from a WKT representation.
<code>ST_AsText(Geometry) → varchar</code>	Returns the WKT representation of a geometry.

Operations

Function	Description
<code>ST_Boundary(Geometry) → Geometry</code>	Returns the closure of the combinatorial boundary of a geometry.
<code>ST_Buffer(Geometry, distance) → Geometry</code>	Returns the geometry that represents all points whose distance from the specified geometry is shorter than or equal to the specified distance.
<code>ST_Difference(Geometry, Geometry) → Geometry</code>	Returns the geometry value that represents the point set difference of the specified geometries.
<code>ST_Envelope(Geometry) → Geometry</code>	Returns the bounding rectangular polygon of a geometry.
<code>ST_ExteriorRing(Geometry) → Geometry</code>	Returns a line string that represents the exterior ring of the input polygon.
<code>ST_Intersection(Geometry, Geometry) → Geometry</code>	Returns the geometry value that represents the point set intersection of two geometries.

Function	Description
<code>ST_SymDifference(Geometry, Geometry) → Geometry</code>	Returns the geometry value that represents the point set symmetric difference of two geometries.

Relationship tests

Function	Description
<code>ST_Contains(Geometry, Geometry) → boolean</code>	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.
<code>ST_Crosses(Geometry, Geometry) → boolean</code>	Returns True if the specified geometries share some, but not all, interior points in common.
<code>ST_Disjoint(Geometry, Geometry) → boolean</code>	Returns True if the specified geometries do not spatially intersect.
<code>ST_Equals(Geometry, Geometry) → boolean</code>	Returns True if the specified geometries represent the same geometry.
<code>ST_Intersects(Geometry, Geometry) → boolean</code>	Returns True if the specified geometries spatially intersect in two dimensions.
<code>ST_Overlaps(Geometry, Geometry) → boolean</code>	Returns True if the specified geometries share space in the same dimension, but are not completely contained by each other.
<code>ST_Relate(Geometry, Geometry) → boolean</code>	Returns True if the first geometry is spatially related to the second geometry.
<code>ST_Touches(Geometry, Geometry) → boolean</code>	Returns True if the specified geometries have at least one point in common, but their interiors do not intersect.
<code>ST_Within(Geometry, Geometry) → boolean</code>	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
<code>ST_Area(Geometry) → double</code>	Returns the two-dimensional Euclidean area of a geometry.
<code>ST_Centroid(Geometry) → Geometry</code>	Returns the point value that is the mathematical centroid of a geometry.
<code>ST_CoordDim(Geometry) → bigint</code>	Returns the coordinate dimension of a geometry.
<code>ST_Dimension(Geometry) → bigint</code>	Returns the inherent dimension of a geometry object, which must be less than or equal to the coordinate dimension.

Function	Description
ST_Distance(Geometry, Geometry) → double	Returns the minimum two-dimensional Cartesian distance (based on spatial ref) between two geometries in projected units.
ST_IsClosed(Geometry) → 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) → 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) → double	Returns the X maximum of the bounding box of the geometry.
ST_YMax(Geometry) → double	Returns the Y maximum of the bounding box of the geometry.
T_XMin(Geometry) → double	Returns the X minimum of the bounding box of the geometry.
ST_YMin(Geometry) → double	Returns the Y minimum of the bounding box of the geometry.
ST_StartPoint(Geometry) → point	Returns the first point of a geometry LineString instance.
ST_EndPoint(Geometry) → point	Returns the last point of a geometry LineString instance.
ST_X(Point) → double	Returns the X coordinate of a point.
ST_Y(Point) → double	Returns the Y coordinate of a point.
ST_NumPoints(Geometry) → bigint	Returns the number of points in a geometry.
ST_NumInteriorRing(Geometry) → bigint	Returns the cardinality of the collection of interior rings of a polygon.

5.4.8.31. 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

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: <code>geohash('34.1,120.6')</code>	<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>

5.4.8.32. 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 *
if __name__=='__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 s._
    _date__ >'2018-04-10 00:00:00' and s.__date__ < '2018-04-11 00:00:00' and m.__date__ >'2018-04-23 00:0
    0:00' and m.__date__ <'2018-04-24 00:00:00' and s.projectid = cast(m.ikey as 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](#).

5.4.8.33. 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-tianchil11932.tc __topic__: TestTopi
c_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
<code>unnest(array) as table_alias(column_name)</code>	Expands an array into multiple rows. <code>column_name</code> specifies the column name of the rows.
<code>unnest(map) as table(key_name, value_name)</code>	Expands a map into multiple rows. <code>key_name</code> specifies the column name of the keys and <code>value_name</code> specifies the column name of the values.

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](#).

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)
```

5.4.9. Machine learning syntax and functions

5.4.9.1. 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

Type	Function	Description
Smooth functions	ts_smooth_simple	Uses the Holt-Winters forecasting algorithm to filter time series data.
	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.
Prediction and anomaly detection functions	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.
	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.
Time series clustering functions	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.
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.

Type	Function	Description
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.

5.4.9.2. 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
<code>ts_smooth_simple</code>	Uses the Holt-Winters forecasting algorithm to filter time series data. This function is the default smooth function.
<code>ts_smooth_fir</code>	Uses a finite impulse response (FIR) filter to filter time series data.
<code>ts_smooth_iir</code>	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.

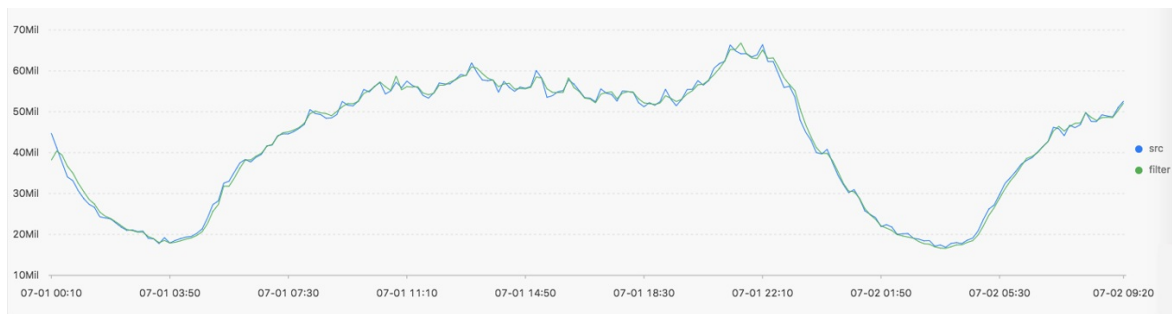
Parameter	Description	Value
<i>x</i>	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at a specific point in time.	None

- 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.
Vertical axis	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
<i>x</i>	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.
<i>y</i>	The sequence of numeric data at a specific point in time.	None
<i>winType</i>	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 Note We recommend that you set the winType parameter to rectangle for better display effects.

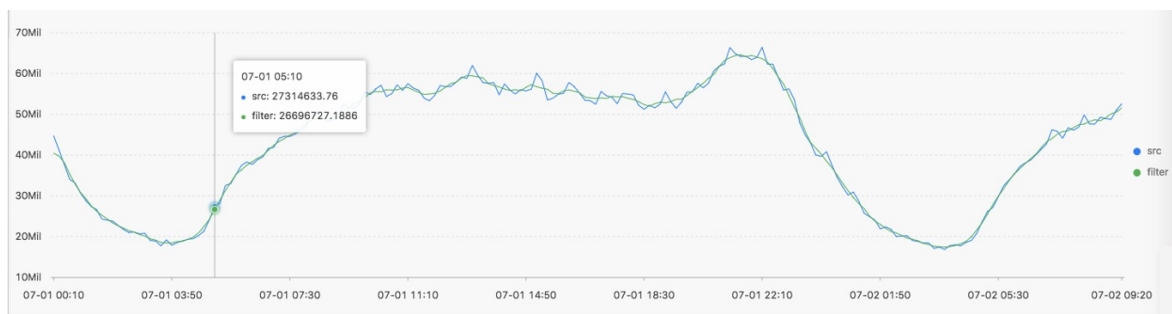
Parameter	Description	Value
<i>winSize</i>	The length of the filter window.	The value is of the long type. Valid values: 2 to 15.
<i>array[]</i>	The parameter that you want to use for FIR filtering.	The value is an array and the sum of the elements in the array is 1. Example: array[0.2, 0.4, 0.3, 0.1].

- 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 )
```

- Query result



- 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 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.
Vertical axis	src	The unfiltered data.
	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.

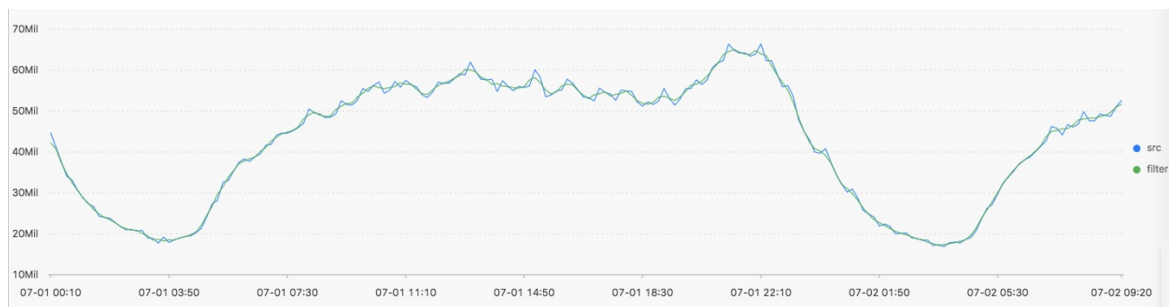
Parameter	Description	Value
<i>x</i>	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at a specific point in time.	None
<i>array[]</i>	The parameter that you want to use for IIR filtering in terms of x_i .	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].
<i>array[]</i>	The parameter that you want to use for IIR filtering in terms of y_{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 )
```

- Query result



- The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	The UNIX timestamp of the data. Unit: seconds.
Vertical axis	src	The unfiltered data.
	filter	The filtered data.

5.4.9.3. 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
<code>ts_period_detect</code>	Estimates the periodicity of time series data that is distributed in different time intervals.
<code>ts_period_classify</code>	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
<i>x</i>	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.
<i>y</i>	The sequence of numeric data at a specific point in time.
<i>minPeriod</i>	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].
<i>maxPeriod</i>	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 stamp order by stamp )
```

- Query result



The following table describes the display items.

Display item	Description
<code>period_id</code>	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.

Display item	Description
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
asg-2zyv6z5ewg18pg5	1.0	-1.0
asg-bp1j8enc92p6v5pptgpj	0.07203669207039314	0.0
asg-wz9fhse7u4ubopo5dt9o	0.0	0.0
asg-bp18oqn10gg96vy85te4	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.

Display item	Description
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.

5.4.9.4. 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
<code>ts_cp_detect</code>	Detects the intervals in which data has different statistical features. The interval endpoints are change points.
<code>ts_breakout_detect</code>	Detects the points in time at which data dramatically changes.

ts_cp_detect

Syntax

```
select ts_cp_detect(x, y, minSize)
```

The following table describes the parameters in the function.

Parameter	Description	Value
<i>x</i>	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.
<i>y</i>	The sequence of numeric data at a specific point in time.	None
<i>minSize</i>	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_cp_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.
Vertical axis	src	The unfiltered data. Example: 1956092.7647745228.
	prob	The probability that a point in time is a change point. Valid values: 0 to 1.

ts_breakout_detect

Syntax

```
select ts_breakout_detect(x, y, winSize)
```

The following table describes the parameters in the function.

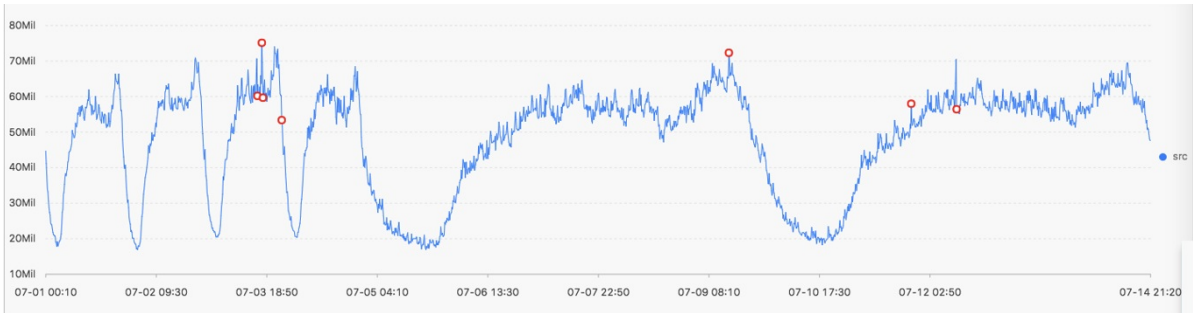
Parameter	Description	Value
<i>x</i>	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at a specific point in time.	None
<i>winSize</i>	The minimum length of time series 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.
	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.

5.4.9.5. 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.

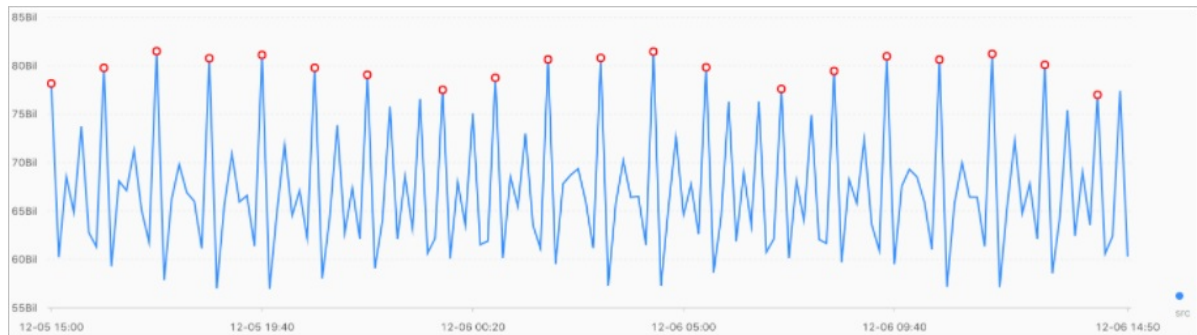
Parameter	Description	Value
<i>x</i>	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at a specific point in time.	None
<i>winSize</i>	The minimum length of the detection 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)
```

- Query result



The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	The timestamp of the data. Unit: seconds. Example: 1537071480.
Vertical axis	src	The unfiltered data. Example: 1956092.7647745228.
	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.

5.4.9.6. 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.

- [Introduction to Log Service machine learning \(01\): time series statistics modeling](#)
- [Introduction to Log Service machine learning \(03\): time series anomaly detection modeling](#)
- [Introduction to Log Service machine learning \(05\): time series prediction](#)
- [Best practices for Log Service machine learning: time series anomaly detection and alerting](#)

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_arma	Uses an autoregressive integrated moving average (ARIMA) model to model time series data, predict time series data, and detect anomalies.

Function	Description
<code>ts_regression_predict</code>	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.
<code>ts_anomaly_filter</code>	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.

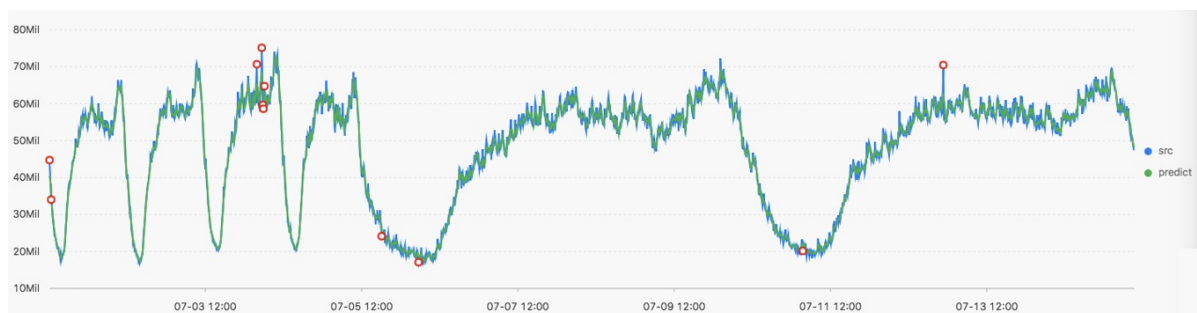
Parameter	Description	Value
<i>x</i>	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at a specific point in time.	None
<i>nPred</i>	The number of points for prediction.	The value is of the long type. This value must be equal to or greater than 1.
<i>isSmooth</i>	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 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 axis	src	The raw data.
	predict	The predicted data.
	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

Syntax

```
select ts_predicate_ar(x, y, p, nPred, isSmooth)
```

The following table describes the parameters in the function.

Parameter	Description	Value
<i>x</i>	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.
<i>y</i>	The sequence of numeric data at a specific point in time.	None
<i>p</i>	The order of the AR model.	The value is of the long type. Valid values: 2 to 8.
<i>nPred</i>	The number of points for prediction.	The value is of the long type. Valid values: 1 to 5 <i>p</i> .
<i>isSmooth</i>	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 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_arma

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.
y	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 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_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.
y	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.

Parameter	Description	Value
<i>nPred</i>	The number of points for prediction.	The value is of the long type. Valid values: 1 to 50.
<i>isSmooth</i>	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)
```

The following table describes the parameters in the function.

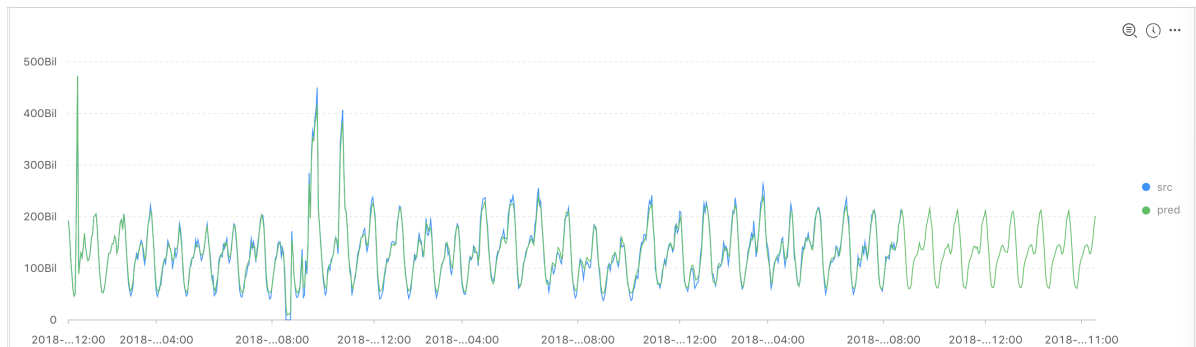
Parameter	Description	Value
<i>x</i>	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at a specific point in time.	None
<i>nPred</i>	The number of points for prediction.	The value is of the long type. Valid values: 1 to 500.
<i>algotype</i>	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 components and obtains the predicted data.	None
<i>processType</i>	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
m (select __time__ - __time__ % 60 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.
Vertical 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
<i>lineName</i>	The name of each curve. The value is of the varchar type.	None
<i>ts</i>	The time sequence of the curve, which indicates the time of the current curve.	The value of this parameter is an array of points in time of the double type. The points in time are sorted in ascending order.
<i>ds</i>	The actual 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.
<i>preds</i>	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.

Parameter	Description	Value
<i>probs</i>	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.
<i>nWatch</i>	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.
<i>anomalyType</i>	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

```
* | 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 lower, 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 | ds | p |
|-----|-----|-----|-----|
| asg-bp1hylzdi2wx7civ0ivk | [1.5513696E9, 1.5513732E9, 1.5513768E9, 1.5513804E9] | [1,2,3,NaN] | [1,2,3,4] | [0,0,1,NaN] |
```

5.4.9.7. 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.

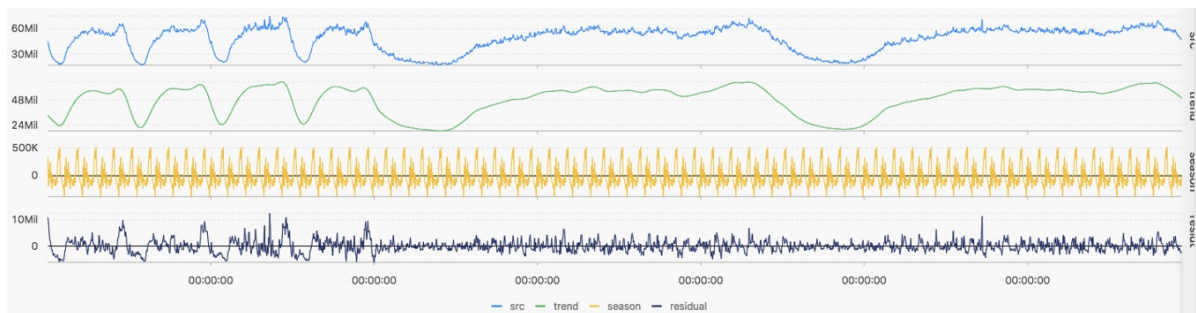
Parameter	Description	Value
x	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
y	The sequence of numeric data at a specific point in time.	None

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



The following table describes the display items.

Display item		Description
Horizontal axis	unixtime	The UNIX timestamp of the data. Unit: seconds.
Vertical axis	src	The raw data.
	trend	The decomposed data that indicates the trend of the time series data.
	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.

5.4.9.8. 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
<code>ts_density_cluster</code>	Uses a density-based clustering method to cluster multiple time series.

Function	Description
<code>ts_hierarchical_cluster</code>	Uses a hierarchical clustering method to cluster multiple time series.
<code>ts_similar_instance</code>	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.

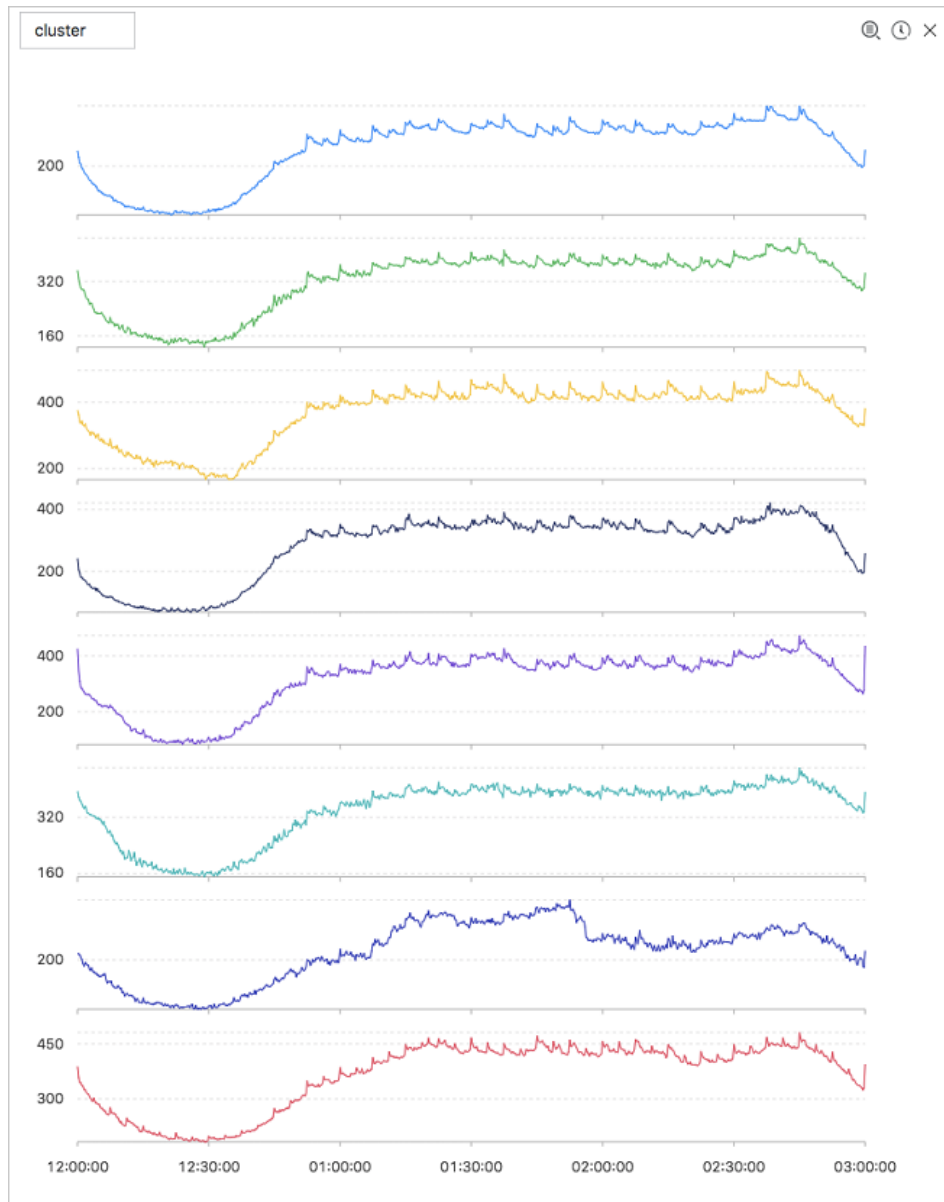
Parameter	Description	Value
<i>x</i>	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at a specific point in time.	None
<i>z</i>	The name of the curve that corresponds 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)
```

The following table describes the parameters in the function.

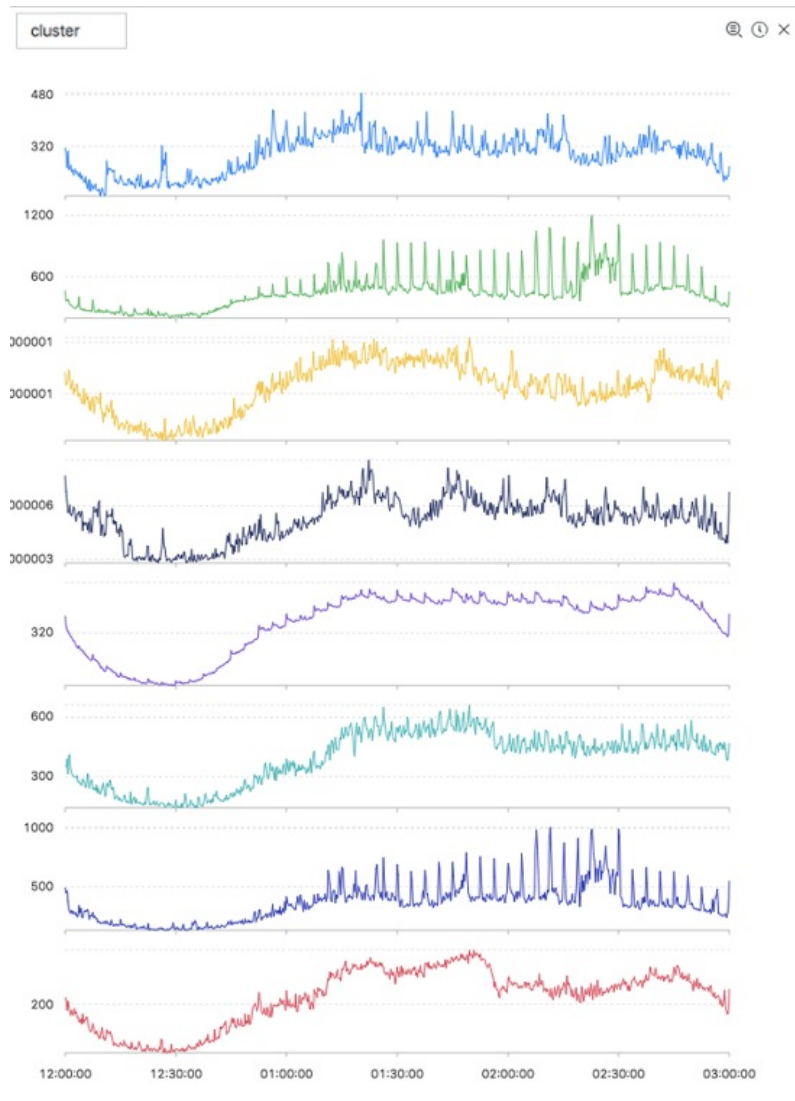
Parameter	Description	Value
<i>x</i>	The time sequence. Points in time are sorted in ascending order along the horizontal axis.	Each point in time is a UNIX timestamp. Unit: seconds.
<i>y</i>	The sequence of numeric data at a specific point in time.	None
<i>z</i>	The name of the curve that corresponds 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)
```

The following table describes the parameters in the function.

Parameter	Description	Value
<i>x</i>	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.
<i>y</i>	The sequence of numeric data at a specific point in time.	None
<i>z</i>	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.
<i>instance_name</i>	The name of a specified curve that you want to query.	The value is of this parameter is a string. Example: machine01.cpu_usr.  Notice You must specify an existing curve.
<i>topk</i>	The maximum number of curves that are similar to the specified curve can be returned.	None
<i>metricType</i>	<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 log GROUP BY stamp, metric_name order BY metric_name, stamp )
```

The following table describes the display items.

Display item	Description
<i>instance_name</i>	The list of metrics that are similar to the specified metric.
<i>time_series</i>	The timestamp sequence of the cluster center.
<i>data_series</i>	The data sequence of the cluster center.

5.4.9.9. 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
<i>array[col1, col2, col3]</i>	A column of character values.	An array of values, for example, array[clientIP, sourceIP, path, logstore].
<i>array['col1_name', 'col2_name', 'col3_name']</i>	The field names of the character values.	An array of field names, for example, array['clientIP', 'sourceIP', 'path', 'logstore'].
<i>array[col5, col6]</i>	A column of numeric values.	An array of values, for example, array[Inflow, OutFlow].
<i>array['col5_name', 'col6_name']</i>	The field names of the numeric values.	An array of field names, for example, array['Inflow', 'OutFlow'].
<i>support_score</i>	The support ratio of samples for pattern mining.	The value is of the DOUBLE data type. Value range: (0,1].
<i>sample_ratio</i>	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.

5.4.9.10. 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 )
```

The following table describes the parameters in the function.

Parameter	Description	Value
<i>array_char_value</i>	A column of values of the character data type.	The value of this parameter is an array. Example: array[clientIP, sourceIP, path, logstore].
<i>array_char_name</i>	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'].
<i>array_numeric_value</i>	A column of numeric values.	The value of this parameter is an array. Example: array[Inflow, OutFlow].
<i>array_numeric_name</i>	The column names of the numeric values.	The value of this parameter is an array. Example: array['Inflow', 'OutFlow'].
<i>condition</i>	The condition that is used to filter data. The value True indicates positive samples and the value False indicates negative samples.	Example: Latency <= 300.
<i>supportScore</i>	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].
<i>posSampleRatio</i>	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].
<i>negSampleRatio</i>	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.

5.4.9.11. 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(vvarchar_array, name_array, real, forecast, level)
```

The following table describes the parameters in the function.

Parameter	Description	Value
<i>vvarchar_array</i>	The array of subdimension attributes.	Example: array[col1, col2, col3].
<i>name_array</i>	The array of subdimension attribute names.	Example: array['col1', 'col2', 'col3'].
<i>real</i>	The actual value of each subdimension attribute that is specified by the vvarchar_array parameter. The value of this parameter is of the double type.	Valid values: all real numbers.
<i>forecast</i>	The predicted value of each subdimension attribute that is specified by the vvarchar_array parameter. The value of this parameter is of the double type.	Valid values: all real numbers.
<i>level</i>	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 vvarchar_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
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)
```

- Query result



The following figure shows the structure of the query result.

```
{
  "rcSets": [
    {
      "rcItems": [
        {
          "kpi": [{"attr": "xxx", "val": "xxx"}],
          "nleaf": 100,
          "change": 0.524543,
          "score": 0.1454543
        }
      ]
    }
  ]
}
```

The following table describes the display items.

Display item	Description
<i>rcSets</i>	The root cause sets. Each value is an array.
<i>rcItems</i>	A root cause set.
<i>kpi</i>	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.
<i>nleaf</i>	The number of leaf nodes that a kpi in the root cause set covers in the raw data. ? Note A leaf node is a log entry that contains the finest-grained attributes.
<i>change</i>	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.
<i>score</i>	The abnormality score of the current kpi. Valid values: [0,1].

The following example shows the query result that is in the JSON format:

```
{
  "rcSets": [
    {
      "rcItems": [
        {
          "kpi": [
            {
              "attr": "country",
              "val": "*"
            },
            {
              "attr": "province",
              "val": "*"
            },
            {
              "attr": "provider",
              "val": "*"
            },
            {
              "attr": "domain",
              "val": "download.huya.com"
            },
            {
              "attr": "method",
              "val": "*"
            }
          ],
          "nleaf": 119,
          "change": 0.3180687806279939,
          "score": 0.14436007709620113
        }
      ]
    }
  ]
}
```

5.4.9.12. 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
<code>ts_association_analysis</code>	Identifies the metrics that are correlated to a specified metric among multiple observed metrics in the system.
<code>ts_similar</code>	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
<i>stamp</i>	The UNIX timestamp that is of the long type.	None
<i>params</i>	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.
<i>names</i>	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.
<i>indexName</i>	The name of the target metric. The value of this parameter is of the varchar type.	Example: Latency.
<i>threshold</i>	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
<i>stamp</i>	The UNIX timestamp that is of the long type.	None

Parameter	Description	Value
<i>value</i>	The value of the metric that you want to analyze. The value of this parameter is of the double type.	None
<i>ts</i>	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
<i>ds</i>	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
<i>metricType</i>	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
<i>stamp</i>	The UNIX timestamp that is of the long type.	None
<i>value</i>	The value of the metric that you want to analyze. The value of this parameter is of the double type.	None
<i>startStamp</i>	The start timestamp of the specified time series curve. The value of this parameter is of the long type.	None
<i>endStamp</i>	The end timestamp of the specified time series curve. The value of this parameter is of the long type.	None
<i>step</i>	The time interval between two adjacent data points in a time series. The value of this parameter is of the long type.	None
<i>ds</i>	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

Parameter	Description	Value
<i>metricType</i>	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 vhost, metric;
```

- Query result

```
| vhost | metric | score |
| ----- | ----- | ----- |
| vhost1 | redolog | -0.3519082537204182 |
| vhost1 | kv_qps | -0.15922168009772697 |
| vhost1 | file_meta_write | NaN |
```

- Description of the query result

score: the correlation between the analyzed metric and the specified time series curve. Valid values: [-1, 1].

5.4.9.13. 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. - epanechnikov: Epanechnikov curve. - gausener: Gaussian curve.

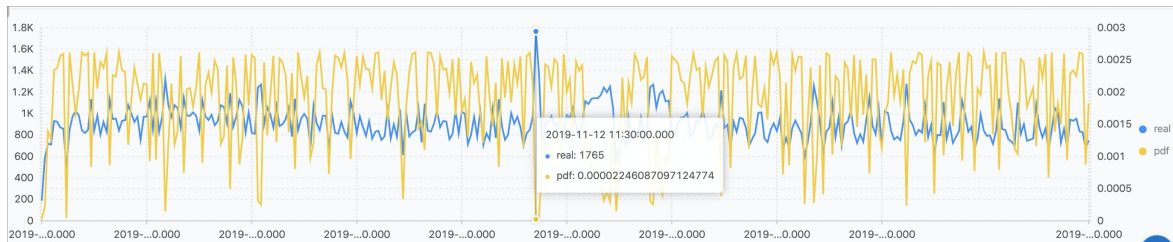
- Response

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



5.4.10. Advanced analysis

5.4.10.1. 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.

- 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 query 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) group by 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')
```

5.4.10.2. 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.

```
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
  FROM      log
  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
```

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.

```
* |
select
  department,
  c * 1.0 / sum(c) over ()
from(
  select
    count(1) as c,
    department
  FROM      log
  group by
    department
)
```

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: `count_if(uri like '%register')` .

5.4.10.3. 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:

- `__time__` : 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 `__time__` 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.host1.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 Pixel/750 Core/UIWebView NetType/WIFI QBWebViewType/1
method: GET
referer: www.host0.com
remote_addr: 36.63.1.23
request_length: 111
request_time: 2.705
status: 200
upstream_response_time: 0.225582883754
url: /?k0=v9&
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

```

5.4.11. Associate Log Service with external data sources

5.4.11.1. 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
 - You can use SQL statements to analyze data and obtain the analysis results within seconds.
 - 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

5.4.11.2. 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.

Context

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`.
 - 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 the region 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 the region parameter to the region where the ECS instance resides. ■ If you use a self-managed MySQL database that is created in other scenarios, set the region 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 the vpc-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 the vpc-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 the instance-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 the instance-id parameter to the ID of the ECS instance. ■ If you use a self-managed MySQL database that is created in other scenarios, set the instance-id parameter to an empty string. Format: "instance-id": "".

Parameter	Description
host	The address of your MySQL database. ■ If you use a MySQL database deployed on an RDS instance that resides in a VPC, set the host 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 the host 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 the host 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 the port 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 the port 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 the port 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's next

[JOIN operations between Logstores and Relational Database Service \(RDS\) tables](#)

5.4.11.3. 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.

```
* | create table user_meta1 ( userid bigint, nick varchar, gender varchar, province varchar, gender varchar, age bigint) with ( endpoint='oss0036-cn-qingdao-env17-d01-a.intra.env17e.shuguang.com', accessid='*****', accesskey = '*****', bucket='testoss****', objects=ARRAY['user.csv'], type='oss')
```

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: <code>user_meta1</code> .
Table schema	The properties of the virtual external table, including the column names and data types. Example: <code>(userid bigint, nick varchar, gender varchar, province varchar, gender varchar, age bigint)</code> .
endpoint	The internal endpoint of OSS. For more information, see Manage a project .
accessid	The AccessKey ID of your account. For more information, see Obtain an AccessKey pair .
accesskey	The AccessKey secret of your account. For more information, see Obtain 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 <code>objects</code> parameter is an array. The array can contain multiple elements. Each element represents an OSS object.

Parameter	Description
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_meta1
```

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. l 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_meta1 u on l.userid = u.userid
```

5.4.11.4. 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'
accessKeyId='test-project'
accessKey='TAI****YDw'
project='lr****VM'
ext_logstore='user_meta'
csv_file='./user.csv'
client = LogClient(endpoint, accessKeyId, accessKey)
res = client.create_external_store(project,
    ExternalStoreCsvConfig(ext_logstore, csv_file,
        [
            {"name" : "userid", "type" : "bigint"},
            {"name" : "nick", "type" : "varchar"},
            {"name" : "gender", "type" : "varchar"},
            {"name" : "province", "type" : "varchar"},
            {"name" : "age", "type" : "bigint"}
        ]))
res.log_print()
```

Parameter	Description
endpoint	The Log Service endpoint. For more information, see Obtain an endpoint in Log Service Developer Guide .
accessKeyId	The AccessKey ID that is used to access Log Service. For more information, see Obtain an AccessKey pair .  Warning 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, see Obtain an AccessKey pair .
project	The project to which the Logstore belongs.

Parameter	Description
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. <pre>[{ "name" : "userid", "type" : "bigint"}, { "name" : "nick", "type" : "varchar"}, { "name" : "gender", "type" : "varchar"}, { "name" : "province", "type" : "varchar"}, { "name" : "age", "type" : "bigint"}]</pre>

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	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	userid	_time_	userid	nick	gender	province	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

5.4.12. Visual analysis

5.4.12.1. Charts

5.4.12.1.1. 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 [Enable the index feature and configure indexes for a Logstore](#).

Note

- Before you configure charts, we recommend that you are familiar with the log analysis feature. For more information, see [Log analysis overview](#).
- 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

To go to the Graph tab, perform the following steps:

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. Click the Logstore that you want to manage.
4. Click **Graph** to go to the Graph tab.

On the **Graph** tab, multiple charts are provided to display query and analysis results. You can select a type of chart from the chart bar based on your business requirements.

- On the **Graph** tab, query and analysis results are displayed in the **Chart Preview** and **Data Preview** sections. The **Chart Preview** section shows the query and analysis results that are displayed in the specified type of chart. The **Data Preview** section shows the data of the related chart in a table.
- On the **Graph** tab, you can configure the following settings:

- On the **Data Source** tab, you can specify placeholder variables. For example, you can configure the drill-down event of Chart A to redirect to the dashboard on which Chart B is located. After you configure the drill-down event of Chart A, the placeholder variable is replaced by the variable that you click to trigger the drill-down event and execute the query statement of Chart B. To trigger the drill-down event, you must click the placeholder variable that you configured for Chart B. For more information, see [Drill-down analysis](#).

This feature is suitable for scenarios in which you want to configure drill-down events to redirect to specified dashboards.

- On the **Properties** tab, you can configure the chart properties that you want to display. You can configure the X-axis, left Y-axis and right Y-axis, margins, and other parameters. Different types of charts have different properties. For more information, see the related topic of each chart.

This feature is suitable for all query and analysis scenarios.

- On the **Interactive Behavior** tab, you can configure drill-down events for a chart. Then, you can click the variable value in the chart to trigger the specified drill-down event. For more information, see [Drill-down analysis](#).

This feature is suitable for scenarios in which you want to trigger drill-down events for charts.

Supported chart types

- Table
- Line chart
- Column chart
- Bar chart
- Pie chart
- Area chart
- Individual value plot
- Progress bar
- Map
- Sankey diagram
- Word cloud
- Treemap chart

5.4.12.1.2. 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

- Log on to the [Log Service console](#).
- 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	Description
Items per Page	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 as <code>__time__</code> and <code>__source__</code> .
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
```

5.4.12.1.3. 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 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 in which the data on the left and right Y-axis is displayed.
Format Right Y-axis	
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 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.

Parameter	Description
Margin	The distance between the axis and the borders of the chart. The parameters include Top Margin, Bottom Margin, Left Margin, and Right Margin.

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.

5.4.12.1.4. 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 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.

5.4.12.1.5. 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 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
```

5.4.12.1.6. 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
- Legend

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 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 pie chart.

Note

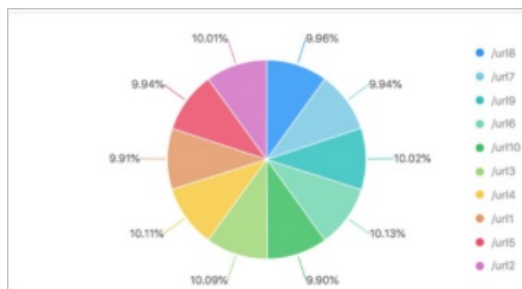
- If a query statement returns no more than 10 log entries, you can use a pie chart or donut chart to display the query results. You can use a `LIMIT` clause to limit the number of segments. If a chart contains a large number of segments with different colors, the analysis results may not be displayed 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:

```
* | select requestURI as uri , count(1) as c group by uri limit 10
```



Example of a donut chart

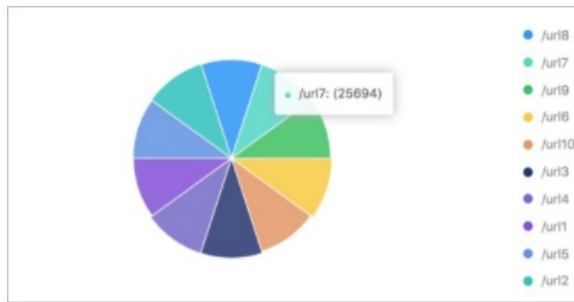
To analyze the percentages of the `requestURI` field values, execute the following query statement:

```
* | 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:

```
* | select requestURI as uri , count(1) as c group by uri limit 10
```



5.4.12.1.7. 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 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.

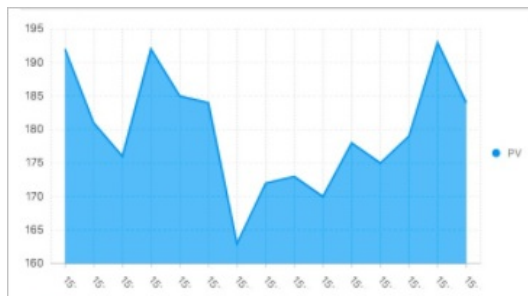
Parameter	Description
X Axis	The sequential data. In most cases, a time series is selected.
Y Axis	The numeric data. You can select one or more fields for the Y-axis.
Legend	The position of the legend in the chart. Valid values: Top, Bottom, Left, and Right.
Format	The format in which data is displayed.
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.

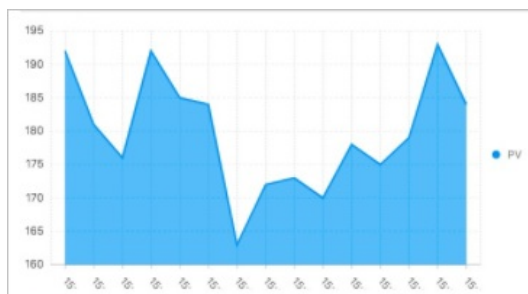


Example of a cascade chart

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 `PV` and `UV` for Y Axis.



5.4.12.1.8. 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 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.

 **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.

- 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.

Parameter	Description
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.  Note By default, a dial is evenly divided into three colored segments. If you change the value of Colored Regions, Region Max Value is not automatically adjusted. You can manually set the maximum value for each colored segment based on your business requirements.
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.

Parameter	Description
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
```

- Compare numb chart

To view and compare the PVs on the current day and in the previous day, execute the following query statement:

```
* | select diff[1],diff[2], diff[1]-diff[2] from (select compare( pv , 86400) as diff from (select count(1) as pv from log))
```

5.4.12.1.9. 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 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.

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.

Parameter	Description
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.  Note The value of Actual Value is compared with the value of Threshold based on the condition specified by Operator. If the actual value matches the condition specified by Operator, the progress bar is displayed in the color specified by Display Color. If the actual value does not match the condition, the progress bar is displayed in the default color.
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))
```

5.4.12.1.10. 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 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  icon..
6. 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:
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 d  
esc limit 10
```

Select *address* for Provinces and *count* for Value Column.

5.4.12.1.11. 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 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 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 .

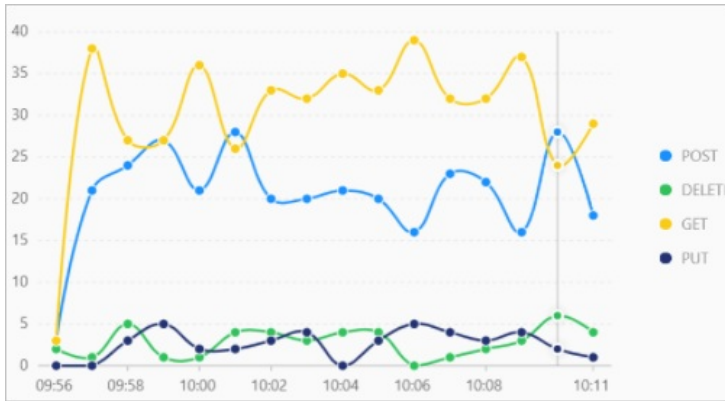
Example

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.

Flow chart



5.4.12.1.12. 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

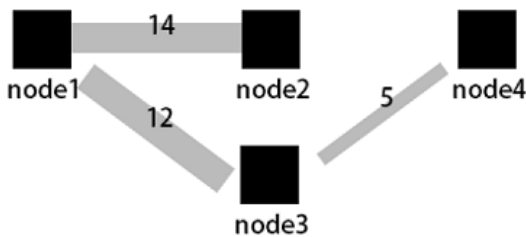
A Sankey diagram has the following features:

- 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
node1	node2	14
node1	node3	12
node3	node4	5

The following figure shows the data relationships in a Sankey diagram.



Procedure

1. [Log on to the 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 `streamValue` 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
```

5.4.12.1.13. 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 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.

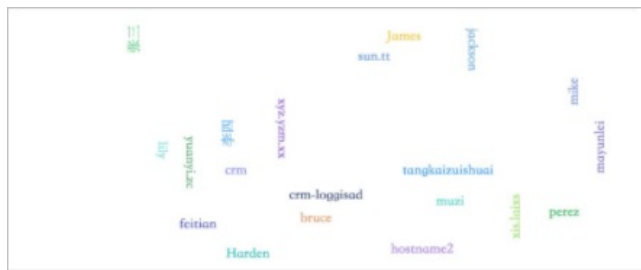
Parameter	Description
Word Column	The words that you want to display.
Value Column	The numeric value that corresponds to a word.
Font Size	The font size of a word. ◦ The minimum font size ranges from 10 pixels to 24 pixels. ◦ The maximum font size ranges from 50 pixels to 80 pixels.

Example

To query the distribution of hostnames in NGINX logs, execute the following query statement:

```
* | 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.



5.4.12.1.14. 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 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.

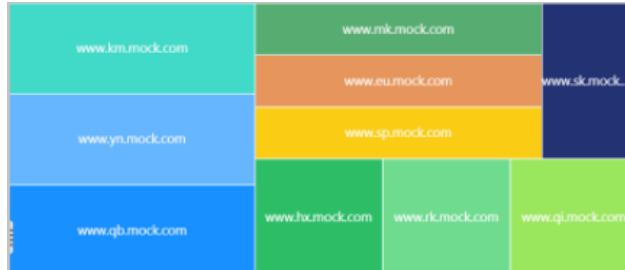
Parameter	Description
Legend Filter	The field that includes categorical data.
Value Column	The numeric value of a field. A greater field value represents a larger rectangle.

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.



5.4.12.2. Dashboard

5.4.12.2.1. 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 [Drill-down analysis](#) 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.

- **Configure the display mode of a dashboard**

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 [Configure and use a filter on a dashboard of a Logstore](#).
- 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.

- **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 [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 [Configure and use a filter on a dashboard of a Logstore](#), 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.

5.4.12.2.2. 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 [Enable the indexing feature and configure indexes for a Logstore](#).

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, 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.



Notice 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.

5.4.12.2.3. 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  icon.
4. In the Dashboard list, click the dashboard that you want to manage.
5. Click **Please Select** to specify a time range.

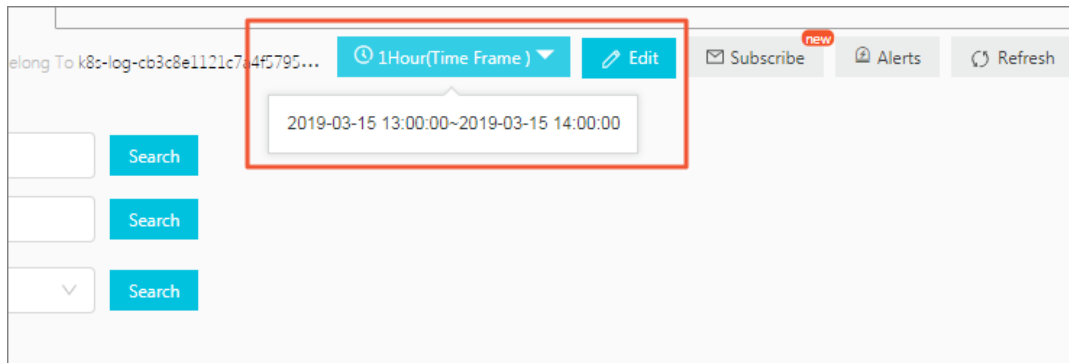
Log Service supports the following types of time ranges:

- **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. Move the pointer over the **Please Select** 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 [Edit mode](#).

Configure an alert rule

In the upper-right corner of the dashboard page, click **Alerts** and select **Create** to create an alert rule for the charts on the dashboard. For more information, see [Configure an alert rule](#).

Configure a refresh method

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.

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  icon and select **Download Log**. 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  icon 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  icon, and then click the  icon. In the Preview Query Statement dialog box, you can view the query statement of the chart.

5.4.12.2.4. 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  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.

 **Notice** 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 [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.
-  **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.
 - You can drag a chart to a position.
 - You can select a chart and set the **L** and **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.
 - 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**.
 - 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 [Drill-down analysis](#).
 - Click **Preview** to check the configuration results.
 - Click **OK**.
 - 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.
 - 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**.
 - Drag the copy to a position. Then, specify the margins and size of the copy.
 - In the upper-right corner of the dashboard page, click **Save**.
- Delete a chart.
 - 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**.
 - In the upper-right corner of the dashboard page, click **Save**.

5.4.12.2.5. 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 drill-down 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.

Context

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.
- **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.
Time Range	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. ■ 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 <code>AND</code> operator.

Parameter	Description
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.  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.

Parameter	Description
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.  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 Value Column: the column in which the variable values are located. The values are used to dynamically replace the variable in the saved search. ■ Static variables ■ Variable: the name of the variable. ■ Value: the fixed value of the static variable. The value is used to replace the placeholder variable in the saved search.

- **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.

Parameter	Description
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. Value: the fixed value of the static variable. The value is synchronized to the dashboard to which you are redirected.

- **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: <code>\${sls_project}</code> , <code>\${sls_dashboard_title}</code> , <code>\${sls_chart_name}</code> , <code>\${sls_chart_title}</code> , <code>\${sls_region}</code> , <code>\${sls_start_time}</code> , <code>\${sls_end_time}</code> , <code>\${sls_realUid}</code> , and <code>\${sls_alid}</code> .
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 dashboard 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.



2. 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 Configurations

request_method [Configure](#) X

Event Action

Open Dashboard

Select Dashboard:

destination_drilldown

Time Range:

Inherit table time

Inherit Filters:

☐

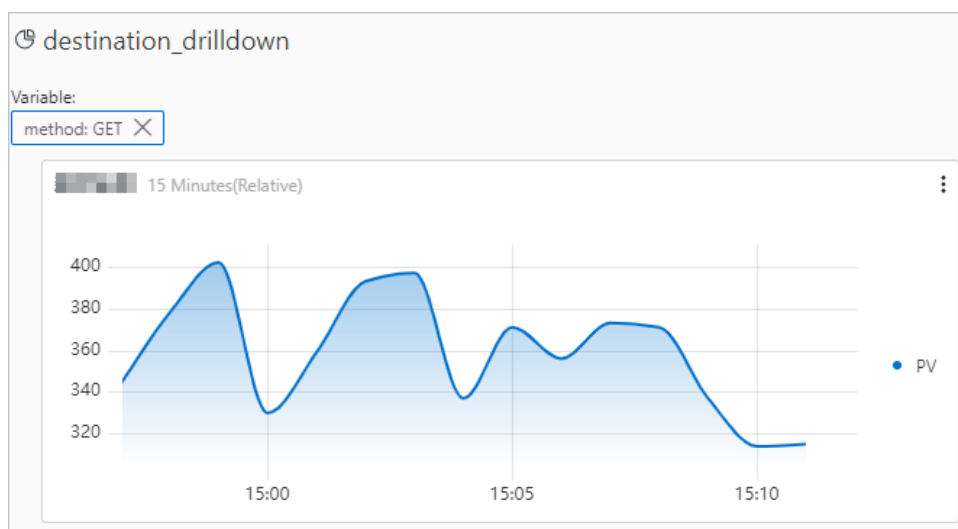
Variable

method

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
GET	5452
POST	1417
PUT	1337
DELETE	562
HEAD	31



5.4.12.2.6. 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 [Enable the indexing feature and configure indexes for a Logstore](#).
- Charts are added to a dashboard. For more information, see [Add a chart to a dashboard](#).

 **Notice** If you set the filter type to Replace Variable, you must configure placeholder variables for the charts on the dashboard.

Context

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

- [Log on to the Log Service console](#).
- Click the name of the project that you want to manage.
- In the left-side navigation pane, click the Dashboard icon.
- In the Dashboard list, click the dashboard that you want to manage.
- In the upper-right corner of the dashboard page, click **Edit**.
- Click the  icon 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.

Parameter	Description
Type	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 the Static 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 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. You can specify multiple values for the variables in the Static List Items field.
Key	◦ If you select Filter, enter the key that you want to use to filter data in the Key field. ◦ If you select Replace Variable, enter the variable that you want to use to filter data in the Key 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 on Global filter. ◦ If you want to filter specific values in specified keys, turn off Global 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 on 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: ◦ Select Logstore: Select a Logstore from which data is queried. ◦ 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.

5.4.12.2.7. Manage a Markdown chart

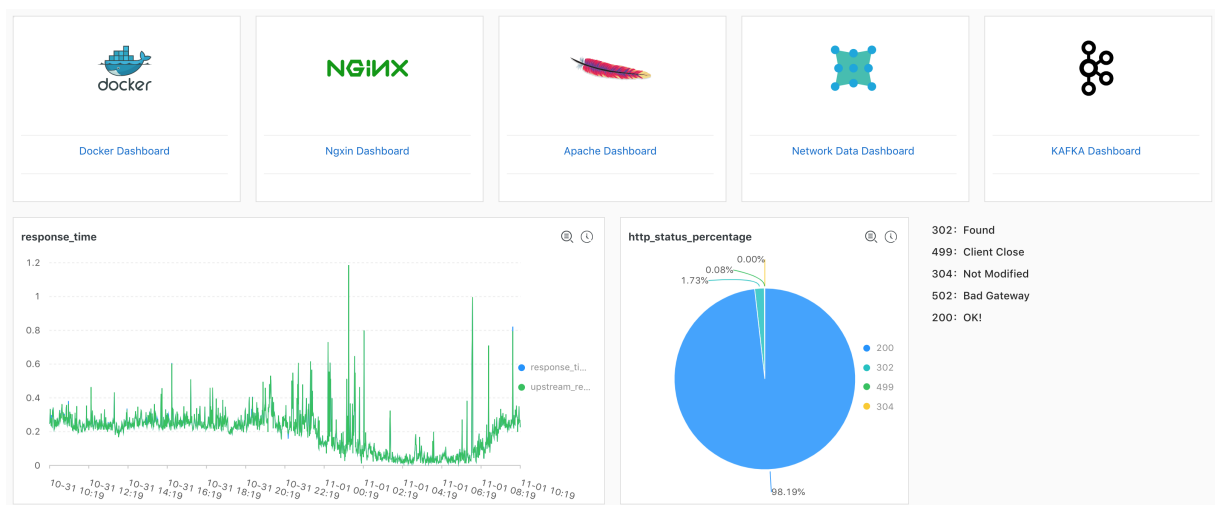
Log Service allows you to add a Markdown chart to a dashboard. In the Markdown chart, you can insert images, links, videos, and other elements to make your dashboard page more intuitive to use.

Context

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 also 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  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  icon 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.

Parameter	Description
Show Title	Specifies whether to show the title of the Markdown chart. You can turn on 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 on Query Binding and associate a query statement with a Markdown chart. Then, dynamic query results are displayed in the Markdown chart. - Select a Logstore whose data you want to query. - Enter a query statement in the search box, specify a time range, and then click Search. For more information, see Overview.  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. - 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**.

2. 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



- Link

Markdown syntax

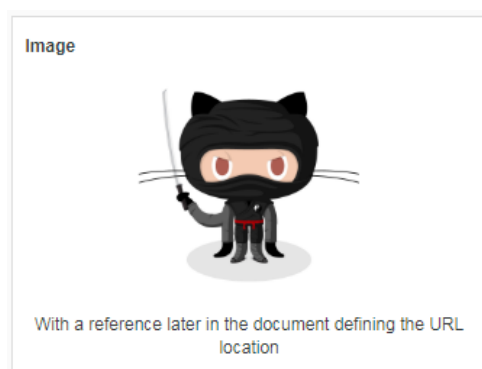
```
### Contents
[Test] (https://www.alibabacloud.com/)
```

- 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

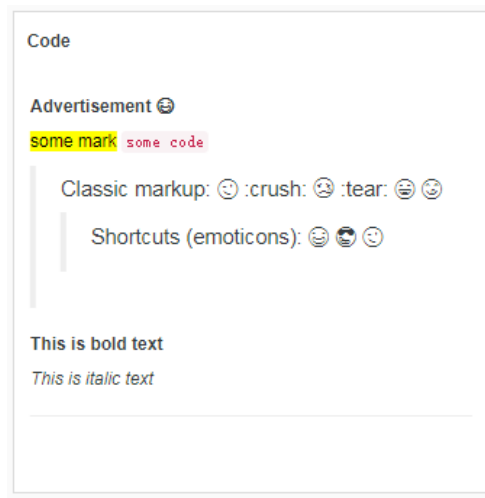


- Special tag

- Markdown syntax

```
---
__Advertisement :)__
==some mark==  `some code`
> Classic markup: :wink: :crush: :cry: :tear: :laughing: :yum:
>> Shortcuts (emojis): :-) 8-) ;)
__This is bold text__
*This is italic text*
---
```

◦ Result



5.4.13. Visualized data analytics

5.5. Alerts

5.5.1. 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.

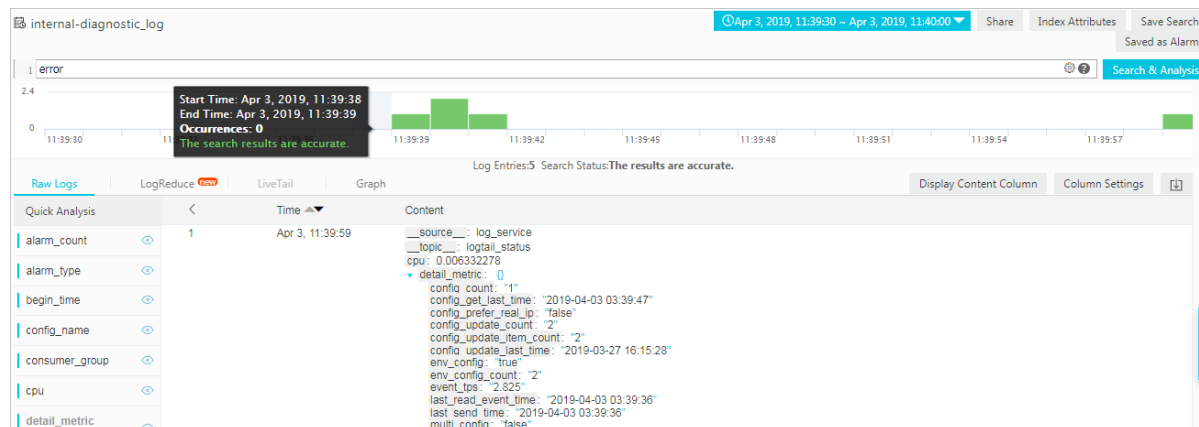
Query statements in alert rules

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.

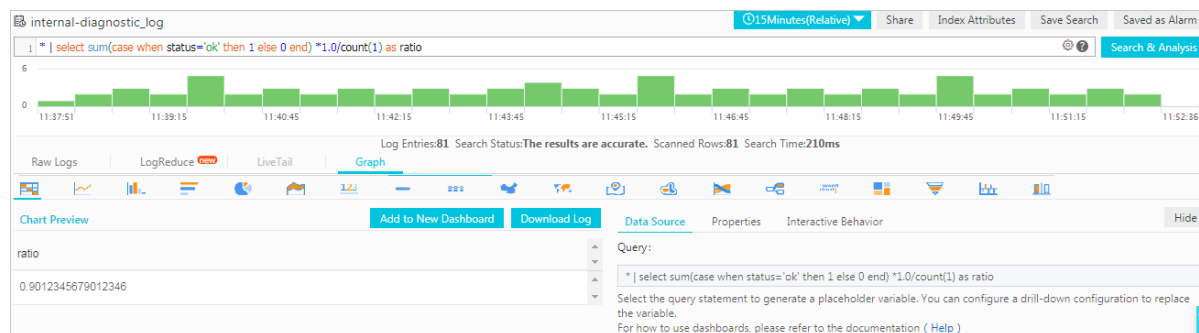
For example, you can execute the error statement to search for the log entries that are generated in the previous 15 minutes and contain error. A total of 154 log entries are returned. Each log entry consists of key-value pairs. You can specify a trigger condition based on the value of a key.

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.



- 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%.



5.5.2. Configure an alarm

5.5.2.1. 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 [Enable the index feature and configure indexes for a Logstore](#).

Context

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.

 **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](#).

For information about common configurations for alert rules, see [FAQ about alerts](#).

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, click the  icon.
4. In the Dashboard list, click the dashboard that you want to manage.
5. In the upper-right corner of the page that appears, choose **Alerts > Create**.
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  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. ◦ If you set the Search Period parameter to 15 Minutes(Time Frame), the time range of the query is 14:15:00-14:30:00.

Parameter	Description
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 <code>pv%100 > 0 && uv > 0</code>.  Note You can use <code>\$serial number</code> to specify a chart in a trigger condition. For example, <code>\$0</code> 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.

- 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 [Notification methods](#).

5.5.2.2. 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.

```
{
  "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.](#)

5.5.2.3. 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.

 **Note** If Log Service does not receive a response within 5 seconds after a notification is sent, the request times out.

1. 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**.

 **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.  Note If you want to remind a group member of the alert notification, use the <code>@<Mobile phone number of the group member></code> syntax in the Content field.

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.

 **Notice** 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 <code>\${Aliuid}</code> .
Project	The project to which an alert rule belongs.	my-project	An alert is triggered in the <code>\${Project}</code> project.
AlertID	The ID of an alert.	0fdd88063a611aa114938f9371daeeb6-1671a52eb23	The ID of the alert is <code>\${AlertID}</code> .
AlertName	The ID of an alert rule. The ID is unique in a project.	alert-1542111415-153472	An alert is triggered based on the <code>\${AlertName}</code> alert rule.

Variable	Description	Example	Reference example
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=AlertID%3A9155ea1ec10167985519fccede4d5fc7-1678293caad&queryTimeType=99&startTime=1544083968	Click \${FullResultUrl} to view the alert details.

Variable	Description	Example	Reference example
Results	The parameters and results of a query. The value is of the array type. For more information, see Fields in alert log entries.  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": [{ "__time__": "1542453580", "count": "0" }], "StartTime": 1542453580 }]</pre>	The first query starts at <code>\${Results[0].StartTime}</code> and ends at <code>\${Results[0].EndTime}</code>. The alert is triggered <code>\${Results[0].FireResult.count}</code> times.  Note In this example, 0 is the serial number of a chart. For more information, see How can I view the serial number of a chart?

5.5.3. Modify and view an alarm

5.5.3.1. 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 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  icon.

4. In the Alerts list, click the alert rule that you want to modify.
5. On the **Alert Overview** page, click **Modify Settings**.
6. 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  icon next to the Query field.

 **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`. You cannot 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**.
8. Modify the evaluation frequency, trigger condition, notification threshold, and notification interval. Then, click **Next**.
For more information, see [Configure alerts](#).
9. Modify the alert notification method and click **Submit**.
For more information, see [Configure alert notification methods](#).

5.5.3.2. 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.

Context

- 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 log entries](#).

 **Note** You are not charged for the internal-alert-history Logstore. The Logstore cannot be deleted.

- 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.

 **Note** The Alert History Statistics dashboard cannot be deleted or modified.

View alert logs in a Logstore

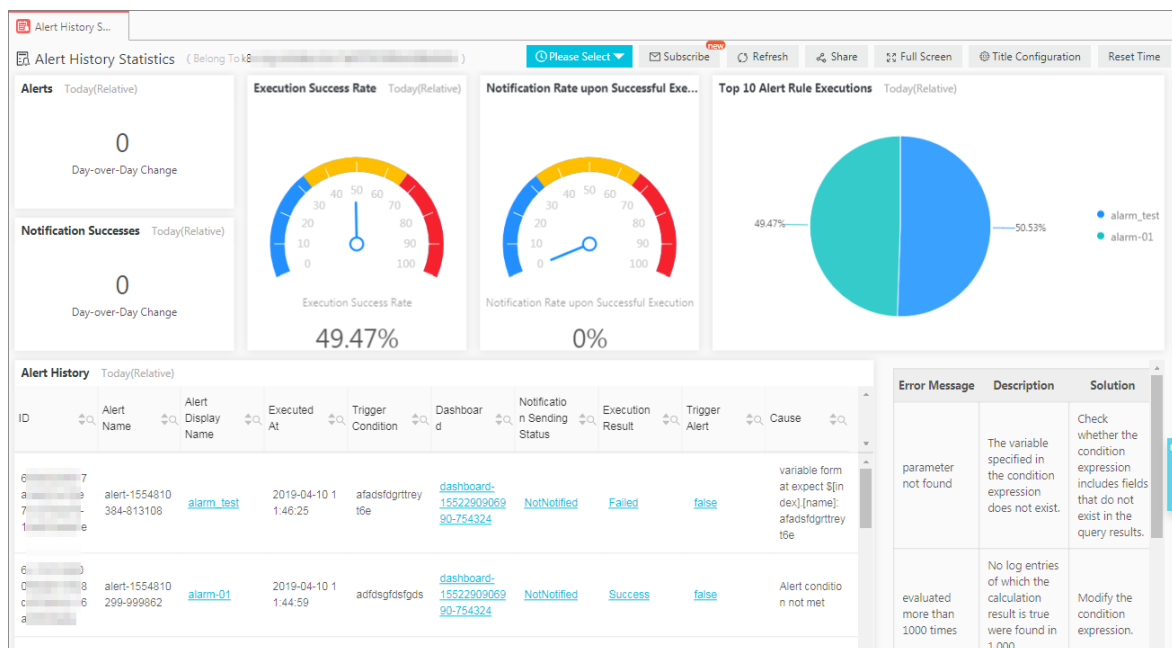
1. [Log on to the Log Service console](#).
2. In the Projects section, click the required project.

- On the **Log Storage > Logstores** tab, find the **internal-alert-history** Logstore and choose  > **Search and Analysis**.
- On the page that appears, query alert logs based on your business requirements.

View the Alert History Statistics dashboard

- Log on to the **Log Service console**.
- In the **Projects** section, click the required project.
- In the left-side navigation pane, click the  icon.
- 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.



5.5.3.3. 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

- Log on to the **Log Service console**.
- In the **Projects** section, click the project that you want to manage.
- In the left-side navigation pane, click the **Alerts** icon.
- In the list of alerts, click the alert whose information you want to view to go to the **Alert Overview** page.
- 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.

Basic Information			
Dashboard		Created At	Jan 19, 2020, 14:57:16
Last Updated At	Jun 23, 2020, 12:49:45	Check Frequency	Cron Expression: 0/5 * * * *
Status	Enabled	Notification Status	Enabled

Disable or enable an alert

After you configure an alert, you can disable or enable the alert at any time.

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**.

Alert Overview (test)				Modify Settings	Delete Alert
Basic Information					
Dashboard		Created At	May 22, 2020, 11:23:35		
Last Updated At	May 22, 2020, 11:37:27	Check Frequency	Fixed Interval 15Minutes		
Status	Enabled	Notification Status	Enabled	Close	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.

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		Created At	Apr 14, 2020, 16:50:53
Last Updated At	Apr 14, 2020, 16:50:53	Check Frequency	Fixed Interval 15Minutes
Status	Enabled	Notification Status	Enabled

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**.

5.5.4. Relevant syntax and fields for reference

5.5.4.1. 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. + - * / %	<code>x * 100 + y > 200</code> <code>x % 10 > 5</code>
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.	<code>x >= 0</code> <code>x < 100</code> <code>x <= 100</code> <code>x == 100</code> <code>x == "foo"</code> - Regex match: <code>x ==~ "\w +"</code>
Logical operators	The AND (&&) and OR () operators are supported.	<code>x >= 0 && y <= 100</code> <code>x > 0 y > 0</code>
Not operator	The not operator (!) is supported.	<code>!(a < 1 && a > 100)</code>
Numeric constants	Numeric constants are supported. Log Service converts numeric constants to 64-bit floating-point numbers.	<code>x > 100</code>
String constants	String constants are supported. The string constants are in the format of 'String', for example, 'string'.	<code>foo == 'string'</code>
Boolean constants	Boolean constants are supported. Valid values: true and false.	<code>(x > 100) == true</code>

Syntax type	Description	Example
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.	<code>x * (y + 100) > 100</code>
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.	<code>contains(foo, 'hello')</code>

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.foo > 100 && \$1.bar < 100, 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.

Operator	Operation method		
	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.

Operator	Operation method		
	Operation between variables	Operation between a non-string constant and a variable	Operation between a string constant and a variable
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 (=~) and regex not match (!~)	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.

Operator	Operation method		
	Operation between variables	Operation between a non-string constant and a variable	Operation between a string constant and a variable
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.

5.5.4.2. 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
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

Log field	Description	Example
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>[{ "EndTime": 1542334900, "FireResult": null, "LogStore": "test-logstore", "Query": "* select count(1) as count", "RawResultCount": 1, "RawResults": [{ "__time__": "1542334840", "count": "0" }], "StartTime": 1542334840 }]</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 <code>[StartTime, EndTime)</code> .	1542334900

Log field	Description	Example
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.	<pre>[{ "__time__": "1542334840", "count": "0" }]</pre>
RawResultsAsKv	The query result that is formatted in key-value pairs.  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.	<pre>{ "__time__": "1542334840", "count": "0" }</pre>
FireResultAsKv	The log entry that records the triggers of an alert. The log entry is formatted in key-value pairs.  Note This field can be used as a template variable. No data is stored to a Logstore for this field.	[foo:0]

5.5.5. FAQ

5.5.5.1. A DingTalk alert notification fails to be sent and the error message "send webhook failed: unable to send request: Post" is reported. Why?

Problem description

A DingTalk alert notification fails to be sent and the error message `send webhook failed: unable to send request: Post ****` is reported.

Possible cause

The DingTalk certificate is not loaded on the physical machine on which Log Service is deployed.

Solution

Download the DingTalk certificate, import the certificate to the physical machine on which Log Service is deployed, and then restart the SlsEtlFramework# server role.

Procedure

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*.

2. Upload the DingTalk certificate to the physical machine on which 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. You can change the name of 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/
```

Information similar to the following example is returned.



- iv. Run the following command to back up the certificate file:

```
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 SlsEtlFramework# 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 Framework**.
- iii. In the left-side navigation pane, click **Project & Cluster O&M**. On the page that appears, enter *sls* in the search box. In the search result, click the SlsEtlFramework# server role.
- iv. On the **Instances** tab, find the instance that you want to use and click **Actions** in the Actions column.
- v. Click **Restart**.

5.6. Real-time consumption

5.6.1. 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](#)

5.6.2. 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);
        PullLogsResponse response = client.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 Preview

internal-alert-history

Shard:0

15 Minutes

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.

Time/Source	Content
2020-05-07 10:40:10	<pre> { "sage": "null", "name": "null", "namespace": "null", "today": "0", "StartTime": 1588818900, "Truncated": false } Reason: Alert condition not met </pre>

5.6.3. Consumption by consumer groups

5.6.3.1. 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 {
    // The endpoint of Log Service. Set the parameter based on your business requirements.
    private static String sEndpoint = "cn-hangzhou.log.aliyuncs.com";
    // 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 consumer.
        // 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, LogHubConfig.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);
    }
}
```

3. Create a file named SampleLogHubProcessor.java.

```

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) {
                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) {
            FastLog log = flg.getLogs(lIdx);
            System.out.println("-----\nLog: " + lIdx + ", time: " + log.getTime() + ", GetC
ontentCount: " + log.getContentsCount());
            for (int cIdx = 0; cIdx < log.getContentsCount(); ++cIdx) {
                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 unexpect
        tedly stops within 30 seconds, a newly started ClientWorker instance continues to consume data fro
        m 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 t
                he server. If you set the parameter to false, 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 c
    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.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.sampleL
ogError(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
.
public LogHubConfig(String consumerGroupName,
                    String consumerName,
                    String loghubEndPoint,
                    String project, String logStore,
                    String accessId, String accessKey,
                    int consumerStartTimeInSeconds);
// 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:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}
log:CreateConsumerGroup	Creates a consumer group for a specified Logstore.	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/*
log:ListConsumerGroup	Queries all consumer groups in a specified Logstore.	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/*

Action	Description	Resource
log:UpdateCheckPoint	Updates the consumption checkpoint in a shard of a specified consumer group.	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/\${consumerGroupName}
log:ConsumerGroupHeartBeat	Sends a heartbeat packet to Log Service for a specified consumer.	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/\${consumerGroupName}
log:UpdateConsumerGroup	Modifies the attributes of a specified consumer group.	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/\${consumerGroupName}
log:ConsumerGroupUpdateCheckpoint	Retrieves the consumption checkpoints in one or all shards of a specified consumer group.	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logstore/\${logstoreName}/consumergroup/\${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:

```
{
  "Version": "1",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "log:GetCursorOrData"
      ],
      "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"
    }
  ]
}
```

5.6.3.2. 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 data in the Logstore";
            else{
                //internal server error
                throw e;
            }
        }
    }
    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).Get
CursorTime();
    }
    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: " + endPr
g);
}
}
}
}

```

5.6.4. Use Storm to consume log data

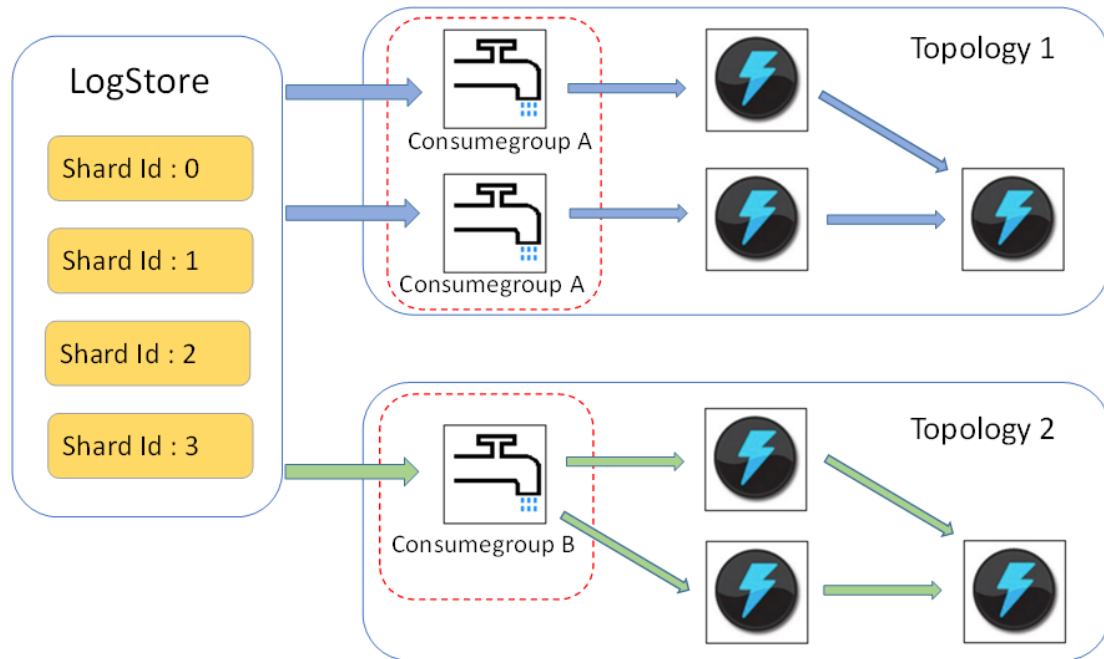
Log Service LogHub provides efficient and reliable log collection and consumption channels. You can use services such as Logtail or SDKs to collect log data in real time. After log data is collected and sent to Log Service, you can use stream computing systems such as Spark Streaming and Apache Storm 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.

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 consumer_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(consumer_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 LogGroupDataSerializSerializer 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();
                // Each 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>
```

5.6.5. 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.

Context

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: <code>com.aliyun.Loghub.flume.sink.LoghubSink</code> .
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 .

Parameter	Required	Description
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 columns. 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: <code>com.aliyun.loghub.flume.source.LoghubSource</code> .
endpoint	Yes	The endpoint of the region where the Log Service project resides.

Parameter	Required	Description
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.  Note If a checkpoint exists on the server, the checkpoint is preferentially used.
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 columns. 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 (").

Parameter	Required	Description
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 named <code>__source__</code> . 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 <code>__tag__: {name of the tag}</code> . 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 named <code>__time__</code> . 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](#).

5.6.6. 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](#).

Context

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>
  <version>0.1.13</version>
</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));

```

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.

2. Specify the start position of log consumption.

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:

- `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-consumer agent uses checkpoints that are stored in the state backend.

3. (Optional) Configure consumption progress monitoring.

The flink-log-consumer 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 group](#).

Example:

```
configProps.put(ConfigConstants.LOG_CONSUMERGROUP, "your consumer group name");
```

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 seconds.
env.enableCheckpointing(5000);
```

For more information, see [Checkpoints](#).

Flink Log Producer

The flink-log-producer agent writes data to Log Service.

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:

```
FlinkLogProducer<String> logProducer = new FlinkLogProducer<String>(new SimpleLogSerializer(),
configProps);
logProducer.setCustomPartitioner(new LogPartitioner<String>() {
    // Generate a 32-bit hash value.
    public String getHashKey(String element) {
        try {
            MessageDigest md = MessageDigest.getInstance("MD5");
            md.update(element.getBytes());
            String hash = new BigInteger(1, md.digest()).toString(16);
            while(hash.length() < 32) hash = "0" + hash;
            return hash;
        } catch (NoSuchAlgorithmException e) {
        }
        return "0000000000000000000000000000000000000000000000000000000000000000";
    }
});
```

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 SimpleLogSerialize
r(), 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 seq = 0;
            while (running) {
                Thread.sleep(10);
                ctx.collect((seq++) + "-" + RandomStringUtils.randomAlphabetic(12));
            }
        }
        @Override
        public void cancel() {
            running = false;
        }
    }
}
```

5.6.7. 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
```

 **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	Type	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 .

Parameter	Type	Required	Description
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 {}
}

```

5.6.8. 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	Type	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.

Parameter	Type	Description
accessKeySecret	String	The AccessKey secret that is used to access Log Service.

- Example

 **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](#).

```
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
object TestLoghub {
  def main(args: Array[String]): Unit = {
    if (args.length < 7) {
      System.err.println(
        """Usage: TestLoghub <project> <logstore> <loghub group name> <endpoint>
          |           <access key id> <access key secret> <batch interval seconds>
        """.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)
    def functionToCreateContext(): StreamingContext = {
      val conf = new SparkConf().setAppName("Test Loghub")
      val ssc = new StreamingContext(conf, batchInterval)
      val loghubStream = LoghubUtils.createStream(
        ssc,
        project,
        logstore,
        consumerGroup,
        endpoint,
        accessKeyId,
        accessKeySecret,
        StorageLevel.MEMORY_AND_DISK)
      loghubStream.checkpoint(batchInterval * 2).foreachRDD(rdd =>
        rdd.map(bytes => new String(bytes)).top(10).foreach(println)
      )
      ssc.checkpoint("hdfs:///tmp/spark/streaming") // set checkpoint directory
      ssc
    }
    val ssc = StreamingContext.getOrCreate("hdfs:///tmp/spark/streaming", functionToCreateContext _
  )
    ssc.start()
    ssc.awaitTermination()
  }
}
```

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	Type	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

Parameter	Description	Default
<code>spark.streaming.loghub.maxRatePerShard</code>	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

```

import 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>
rt=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](#).

5.6.9. 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 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'
);
```

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 <i>Log Service Developer Guide</i> .
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.

Parameter	Required	Description
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.  Note If the size of a single log entry and the number of log groups in a batch are large, the Java system may frequently recycle the data that is stored in the memory.
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
<code>__source__</code>	The source of the log entry.
<code>__topic__</code>	The topic of the log entry.
<code>__timestamp__</code>	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.

 **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.

5.7. Data shipping

5.7.1. Ship logs to OSS

5.7.1.1. 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](#).

5.7.1.2. 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 bucket**s section in the *Service User Guide - Object 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](#).

Context

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.

Procedure

- [Log on to the Log Service console](#).
- 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)**.
- On the OSS Shipper page, click **Enable**.
- In the **OSS LogShipper** pane, configure the shipping rules.

The following table describes the required parameters.

Parameter	Description
OSS Shipper Name	The name of the shipping rule. The name can contain only lowercase letters, digits, hyphens (-), and underscores (_). The name must start and end with a lowercase letter or digit and must be 2 to 128 characters in length.
OSS Bucket	The name of the OSS bucket to which you want to ship log data.  Notice 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.

Parameter	Description
(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: <code>acs:ram::45643:role/aliyunlogdefaultrole</code> . 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 - Object 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, 2016.

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	<code>oss://test-bucket/test-table/2017/01/20/19/50_1484913043351525351_2850008</code>
test-bucket	log_ship_oss_example	year=%Y/mon=%m/day=%d/log_%H%M%s	<code>oss://test-bucket/log_ship_oss_example/year=2017/mon=01/day=20/log_195043_1484913043351525351_2850008.parquet</code>
test-bucket	log_ship_oss_example	ds=%Y%m%d/%H	<code>oss://test-bucket/log_ship_oss_example/ds=20170120/19_1484913043351525351_2850008.snappy</code>
test-bucket	log_ship_oss_example	%Y%m%d/	<code>oss://test-bucket/log_ship_oss_example/20170120/_1484913043351525351_2850008</code> 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	<code>oss://test-bucket/log_ship_oss_example/2017012019_1484913043351525351_2850008</code>

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://test-bucket/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 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.

5.7.1.3. 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 RAM Role**.
4. In the **Organization Name** drop-down list, select the organization that you created. In the **Service Name** drop-down list, select **Log Service**, and click **OK**.
5. On the **RAM Service Role** page, enter `AliyunLogDefaultRole` in the **Role Name** search box and click **Search**.
6. 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.

5.7.1.4. 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_1453812893059571256_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: <pre>{ "__time__": 1453809242, "__to_pic__": "", "__source__": "10.170.***.***", "ip": "10.200.***.***", "time": "26/Jan/2016:19:54:02 +0800", "url": "POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=<yourSignature>" }</pre> HTTP/1.1, "status": "200", "user-agent": "aliyun-sdk-java"
snappy	.snappy	oss://oss-shipper-shenzhen/ecs_test/2016/01/26/20/54_1453812893059571256_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 <code>__time__</code>, <code>__topic__</code>, and <code>__source__</code>.  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_1453812893059571256_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_1453812893059571256_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

Parameter	Description
Key Name	The name of the log field that you want to ship to OSS. You can view log fields on the Raw 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 <code>__time__</code>, <code>__topic__</code>, and <code>__source__</code>. 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.  Note The keys that you enter in the Parquet Keys field must be unique.
Type	The Parquet storage format supports six data types: string, Boolean, Int32, Int64, float, and double. 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_1453812893059571256_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_1453812893059571256_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 .

5.7.1.5. 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](#).
2. Run the decompression script.

The following example is a sample decompression script:

```
import snappy
compressed = open('/tmp/temp.snappy').read()
snappy.uncompress(compressed)
```

 **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
```

5.8. Time series storage

5.8.1. Data import

5.8.1.1. 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 in Linux](#).

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 in Linux](#).

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 in Linux](#).

- 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 a machine group based on a server IP address](#) or [Create a custom ID-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**.

 **Notice** 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.

```
{
  "inputs": [
    {
      "detail": {
        "IntervalMs": 30000
      },
      "type": "metric_system_v2"
    }
  ]
}
```

Parameter	Type	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

Metric	Description	Unit	Example
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

Metric	Description	Unit	Example
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
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's 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.

5.8.1.2. Import metrics collected by Telegraf

5.8.1.2.1. 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:

- 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.

5.8.1.2.2. 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 in Linux](#).
- 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 **MySQL 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 in Linux](#).

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 in Linux](#).

- 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 a machine group based on a server IP address](#) or [Create a custom ID-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**.

 **Notice** 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.

Parameter	Description
Cluster Name	Enter the name of the MySQL cluster. After you configure this parameter, Log Service adds a <code>cluster=<Cluster name></code> tag to the data that is collected.  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.

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's 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 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.

5.8.1.2.3. 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

- Logtail V0.16.48 or later is installed on a Linux server. For more information, see [Install Logtail in Linux](#).
- 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**.
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 in Linux](#).
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 in Linux](#).

- 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 a machine group based on a server IP address](#) or [Create a custom ID-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**.

 **Notice** If you enable a machine group immediately after you create the machine group, the heart beat 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 heart beats 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.

Parameter	Description
Application Name	Enter the name of your Java application. After you configure this parameter, Log Service adds a <code>cluster=<Application name></code> tag to the data that is collected. Note Make sure that the name of the application is unique. Otherwise, data conflicts may occur.
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 use **Jolokia** 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 start up 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](#).

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.

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 response
{"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-jvm","debug":"false","agentType":"jvm","policyLocation":"classpath:/jolokia-access.xml","agentContext":"/jolokia","serializeException":"false","mimeType":"text/plain","maxDepth":"15","authMode":"basic","authMatch":"any","discoveryEnabled":"true","streaming":"true","canonicalNaming":"true","historyMaxEntries":"10","allowErrorDetails":"true","allowDnsReverseLookup":"true","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's 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 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.

5.8.1.2.4. 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 Logtail 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 in Linux](#).

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_stub_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 in Linux](#).

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 in Linux](#).

- 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 a machine group based on a server IP address](#) or [Create a custom ID-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**.

 **Notice** 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 <code>cluster=<Cluster name></code> tag to the data that is collected.  Note Make sure that the cluster name is unique. Otherwise, data conflicts may occur.
Server List	Click the + icon to add the NGINX server and configure the following parameters: ◦ Address: the endpoint of the NGINX server. ◦ Port: the port number of the NGINX server. ◦ Path: the URI of the NGINX status module. Example: <code>/private/nginx_status</code>. 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.

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's 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 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.

5.8.1.3. Collect metric data from Prometheus

5.8.1.3.1. 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/v1/write
basic_auth:
  username: access-key-id
  password: access-key-secret
queue_config:
  batch_send_deadline: 20s
  capacity: 20480
  max_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: <code>https://{project}.{sls-endpoint}/prometheus/{project}/{metricstore}/api/v1/write</code>. Take note of the following variables: ◦ <code>{sls-endpoint}</code>: the Log Service endpoint. For more information, see <i>Obtain an endpoint</i> in Log Service Developer Guide. ◦ <code>{project}</code>: the project that you created. ◦ <code>{metricstore}</code>: the Metricstore that you created.  Notice ◦ 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.

Parameter	Description
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	queue_config specifies the cache and retry policies for data writes. To reduce invalid network requests, set min_backoff to a value that is greater than or equal to 100ms and set max_backoff to a value that is greater than or equal to 5s. If you want to collect a large amount of data from Prometheus, use the following settings for queue_config: <pre>batch_send_deadline: 20s capacity: 20480 max_backoff: 5s max_samples_per_send: 2048 min_backoff: 100ms min_shards: 100</pre>

3. 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.

Consumption Preview

✕

te

Shard:0

15 Minutes

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.

Time/Source	Content
2020-10-30 14:00:00 172.16.100.100	__labels__ hostname\$Richao-test(p123456) m_bsd __name__ sysfs m_load1 __time_nano__ 1601447906100000000 __value__ 0.15
2020-10-30 14:00:00 172.16.100.100	__labels__ hostname\$Richao-test(p123456) m_bsd __name__ sysfs m_load5 __time_nano__ 1601447906100000000 __value__ 0.12
2020-10-30 14:00:00 172.16.100.100	__labels__ hostname\$Richao-test(p123456) m_bsd __name__ sysfs m_load15 __time_nano__ 1601447906100000000 __value__ 0.09
2020-10-30 14:00:00 172.16.100.100	__labels__ hostname\$Richao-test(p123456) m_bsd __name__ sysfs m_bsd_time __time_nano__ 1601447906100000000 __value__ 1.587960167e+09
2020-10-30 14:00:00 172.16.100.100	__labels__ hostname\$Richao-test(p123456) m_bsd __name__ cpu_curl __time_nano__ 1601447906100000000 __value__ 1

Close

What's 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](#).

5.8.1.3.2. 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

 **Notice** A Logtail plug-in supports only one Logtail configuration for Prometheus. If more than one configuration exists, Logtail 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.

 **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 in Linux](#).

- 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 a machine group based on a server IP address](#) and [Create a custom ID-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**.

 **Notice** 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.

 Notice

- 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:

```
{
  "processors": [
    {
      "type": "processor_appender",
      "detail": {
        "Key": "__labels__",
        "Value": "|host##{{__host__}}|ip##{{__ip__}}",
        "SortLabels": true
      }
    }
  ]
}
```

For more information, see [Add log fields](#).

What's 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 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 Grafana](#).

5.8.2. Query and analysis

5.8.2.1. 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_name}` specifies the name of the Metricstore that you created.

 **Note** You 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 *Metric*.

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.

Notice

- 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
<code>promql_query(string)</code>	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 <code>/query API</code> of Prometheus. Parameter setting: <code>query=<string></code> .	<pre>* SELECT promql_query('up') FROM metrics</pre>

Function	Description	Example
<code>promql_query_range(string, string)</code>	Evaluates a query on the data over a specified period of time. This function is equivalent to the <code>/query_range</code> API of Prometheus. Parameter settings: <code>query=<string></code> and <code>step=<duration></code> .	<pre>* SELECT promql_query_range('up', '5m') FROM metrics</pre>
<code>promql_labels()</code>	Returns all label keys.	<pre>* SELECT promql_labels() FROM metrics</pre>
<code>promql_label_values(string)</code>	Returns the values of a label.	<pre>* SELECT promql_label_values('__name__') FROM metrics</pre>
<code>promql_series(string)</code>	Returns the time series that is matched.	<pre>* SELECT promql_series('up') FROM metrics</pre>

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 `promql_query_range(string, string)` 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>	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	Type	Description
series	map<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 *Metric*.

5.8.2.2. 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:

- Enter a query statement in the field next to the Metricstore name and click **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 `${{Variable name|Default value}}`. For example, you can set `host=~"^.*"` to `host=~"${{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=~".*"}[1m]))') from metrics limit 10000
```



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  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, see Configure a time series chart .
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 Configure an alert rule .
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, choose Refresh > Once to refresh the time series data in the Metricstore. - In the upper-right corner of the query page, choose Refresh > 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 .

5.8.3. Visualization

5.8.3.1. 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.

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#$#hostname|ip#$#192.0.2.0 __time_nano__:1644309671000000000 __value__:52.71 __name__:cpu_util` . 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.

5.8.3.2. 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.
HTTP	◦ URL: Enter the URL of the Metricstore in the <code>https://{project}.{sls-endpoint}/prometheus/{project}/{metricstore}</code> format. Replace <code>{sls-endpoint}</code> with the Log Service endpoint in the region where the project resides. To obtain the list of Log Service endpoints in different regions, see <i>Obtain an endpoint</i> in Log Service Developer Guide. Replace <code>{project}</code> and <code>{metricstore}</code> with the actual project name and Metricstore name. Example: <code>https://sls-prometheus-test.cn-hangzhou.log.aliyuncs.com/prometheus/sls-prometheus-test/prometheus</code>.  Note To ensure the security of transmissions, use <code>https</code> . ◦ 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	<pre>GET /api/v1/query POST /api/v1/query</pre>
Range queries	<pre>GET /api/v1/query_range POST /api/v1/query_range</pre>
Getting label names	<pre>GET /api/v1/labels POST /api/v1/labels</pre>
Querying label values	<pre>GET /api/v1/label/<label_name>/values</pre>
Finding series by label matchers	<pre>GET /api/v1/series POST /api/v1/series</pre>

5.9. RAM

5.9.1. Overview

Resource Access Management (RAM) is a resource access control service provided by Apsara Stack.

You can use RAM to manage users, including employees, systems, and applications. You can also use RAM to grant users permissions to access resources.

RAM provides the following features:

- RAM Role

To authorize a cloud service in a level-1 organization to use other resources in the organization, you must create a RAM role. This role specifies the operations that the cloud service can perform on the resources.

Only system administrators and level-1 organization administrators can create RAM roles.

- User group

You can create multiple RAM users for an organization and grant the users different permissions on the same cloud resources in the organization.

You can create RAM user groups to classify and authorize RAM users within your Apsara Stack tenant account. This simplifies the management of RAM users and their permissions.

You can create RAM policies to grant permissions to different user groups.

5.9.2. Create a RAM role

To authorize a cloud service in a level-1 organization to use other resources in the organization, you must create a RAM role. This RAM role specifies the operations that the cloud service can perform on the resources.

Procedure

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 of the **Enterprise** page, click **Roles**.
4. In the upper-right corner of the page, click **Create RAM Role**.
5. On the **Roles - Create RAM Role** page, configure the **Role Name** and **Description** parameters.
6. Click **Create**.

5.9.3. Create a user

You can create a user and assign the user different roles as an administrator to meet different requirements for system access control.

Procedure

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 of the **Enterprise** page, click **Users**.
4. On the **Users** page, click **Create a user**.
5. In the **Create a user** dialog box, configure the parameters. The following table describes the parameters.

Parameter	Description
User name	The username.
Display name	The display name of the user.
Role	The roles that you want to assign to the user.
Organization	The organization to which the user belongs.
Logon policy	The logon policy that restricts the logon time and IP address of the user. If you do not configure this parameter, the default policy is attached to the created user.
Phone	The mobile phone number of the user. If you want to send text messages about the resource requests and usage to the mobile phone number, make sure that the specified mobile phone number is valid.
Landline	Optional. The landline number of the user.
Email	The email address of the user. If you want to send emails about the resource requests and usage to the email address, make sure that the specified email address is valid.
DingTalk Key	Optional. The DingTalk key.

Parameter	Description
Notify User by Email	If you select Notify User by Email , emails about the resource requests and usage are sent to the specified email address.
Notify User by DingTalk	If you select Notify User by DingTalk , messages about the resource requests and usage are sent to the specified DingTalk user.

6. Click **OK**.

5.9.4. Create a user group

You can create a user group in a selected organization and grant permissions to the users within the group at the same time.

Prerequisites

An organization is created.

Context

Relationships between user groups and users:

- A user group can contain zero or more users.
- A user does not need to belong to a user group.
- A user can be added to multiple user groups.

Relationships between user groups and organizations:

- A user group belongs to only one organization.
- You can create multiple user groups within an organization.

Relationships between user groups and roles:

- Only one role can be assigned to each user group.
- A role can be assigned to multiple user groups.
- When a role is assigned to a user group, the permissions that the role has are automatically granted to the users within the user group.

Relationships between user groups and resource sets:

- Zero or more user groups can be added to a resource set.
- A user group can be added to multiple resource sets.

Procedure

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 of the **Enterprise** page, click **User Groups**.
4. In the upper-right corner of the page, click **Create a user group**.
5. In the dialog box that appears, configure the **User Group Name**, **Organization**, and **Role authorization** parameters.
6. Click **OK**.

5.9.5. Add a user to a user group

You can add a user to a user group.

Procedure

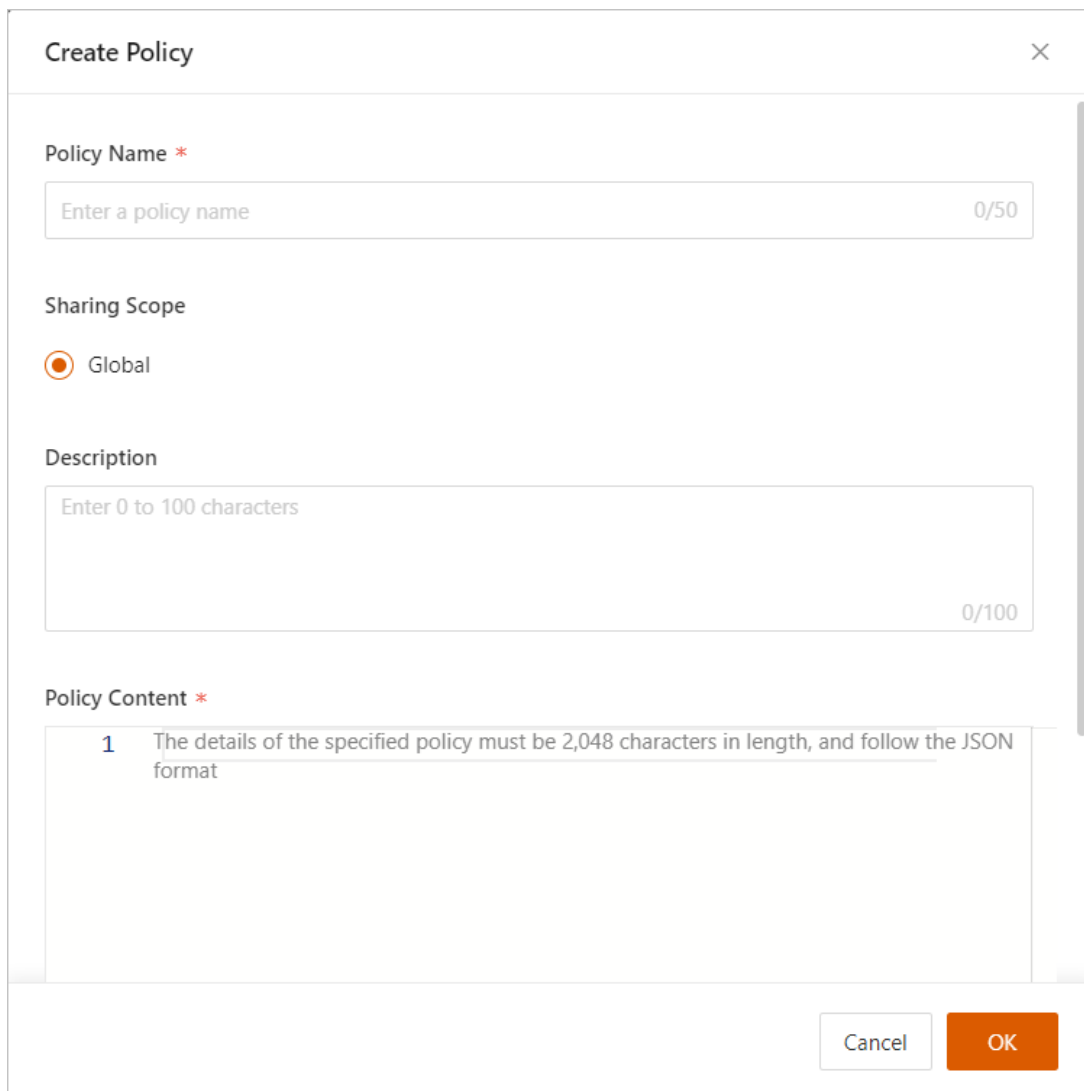
1. Log on to the Apsara Uni-manager Management Console.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **User Groups**.
4. Find the user group to which you want to add a user, and click **Add User** in the **Actions** column.
5. In the dialog box that appears, select the user that you want to add from the left section, and click the right arrow to move the user to the right section.
6. Click **OK**.

5.9.6. Create a policy

To use a cloud service to access other cloud resources, you must create a policy and attach it to a user group.

Procedure

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 of the **Enterprise** page, click **Roles**.
4. In the role name list, find the RAM role that you want to modify and choose **More > Modify** in the **Actions** column to go to the **Roles** page.
5. Click the **Permissions** tab.
6. Click **Add Permission Policy**.
7. In the Add Permission Policy dialog box, enter the information about the policy.



The image shows a 'Create Policy' dialog box with a close button (X) in the top right corner. It contains four main sections: 'Policy Name' with a text input field and a 0/50 character count; 'Sharing Scope' with a radio button selected for 'Global'; 'Description' with a text area and a 0/100 character count; and 'Policy Content' with a text area and a 0/2048 character count. A numbered list item '1' is visible in the 'Policy Content' section, stating: 'The details of the specified policy must be 2,048 characters in length, and follow the JSON format'. At the bottom right, there are 'Cancel' and 'OK' buttons.

Create Policy ×

Policy Name *

Enter a policy name 0/50

Sharing Scope

☒ Global

Description

Enter 0 to 100 characters 0/100

Policy Content *

1 The details of the specified policy must be 2,048 characters in length, and follow the JSON format 0/2048

Cancel OK

5.9.7. Grant a RAM user the permissions to manage a project

This topic describes how to grant a Resource Access Management (RAM) user the permissions to manage a specified project.

Prerequisites

- A RAM user is created. For more information, see [Create a user](#).
- A resource set is created and the RAM user is added to the resource set. For more information, see **Enterprise Center > Resource Sets** in *Apsara Uni-manager Management Console User Guide*.

Procedure

- 1.
2. In the top navigation bar, click **Enterprise**.
3. In the left-side navigation pane of the **Enterprise** page, click **Data Permissions**.
4. Click the resource set that you want to manage, for example, ResourceSet(appstreaming).
5. In the **Product Type** section, click **Log Service**.

6. Find the instance that you want to manage and click **Authorize** in the Actions column.
In this example, the instance is a Log Service project whose name is test-project.
7. Grant the RAM user the permissions to manage the project.
If you turn on the Action switch of the **Update Project** permission, the RAM user can modify the project that is selected in Step .

5.9.8. Grant permissions to a RAM role

When you grant permissions to a RAM role, all users in the user groups that are assigned this role share the granted permissions.

Procedure

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 of the **Enterprise** page, click **Roles**.
4. In the role name list, find the RAM role that you want to modify and choose **More > Modify** in the **Actions** column to go to the **Roles** page.
5. Click the **Permissions** tab.
6. Click **Select Existing Permission Policy**.
7. In the dialog box that appears, select a RAM policy and click **OK**.
If no RAM policies are available, see [Create a permission policy](#).

5.9.9. 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.

Context

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:

- 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:

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": ["log:ListProject"],
      "Resource": ["acs:log:*:*:project/*"],
      "Effect": "Allow"
    },
    {
      "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>"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "log:List*"
      ],
      "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:

- 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*"
      ],
      "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/*"
      ],
      "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:

```
{
  "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 Logstore, use the following policy:

```
{
  "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/logstore/Logstore name",
        "acs:log:*:*:project/Project name/logstore/Logstore name/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

5.10. Monitor Log Service

5.10.1. 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, see Log 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.

Default configuration item	Description
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 <i>Manage shards</i> in <i>Log Service User Guide</i> .
Log retention period	By default, log data is retained for 30 days. You can change the retention period. For more information, see <i>Manage a Logstore</i> in <i>Log Service User Guide</i> .
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.

5.10.2. 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.
Log Storage Location	◦ Automatic creation (recommended): Log Service automatically creates a project named <code>log-service-{User ID}-{region}</code> in the same region to store service logs. We recommend that you store all service logs of the same region in this project. ◦ 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*.

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 **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"
      ],
      "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:*"
      ],
      "Resource": "acs:log:*:*:project/{The project for which the service log feature is enabled}/logging",
      "Effect": "Allow"
    }
  ]
}
```

5.10.3. 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

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.
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	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 specify <code>__topic__: consumergroup_log</code> 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 <code>__topic__: logtail_alarm</code> in the query statement.
			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 Logtail collection logs, you must specify <code>__topic__: logtail_profile</code> in 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 specify <code>__topic__: logtail_status</code> in the query statement.



Notice 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

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: • PutLogs • PutWebtracking
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.	LT AI4FkSqNGBsVTqVZYx****
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

Field	Description	Example
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.	OK
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
Topic	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

Field	Description	Example
Topic	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-23xeb15ni.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

Logtail collection logs are classified into the following categories based on the `file_name` field:

- Statistics about a Logtail configuration for a log file.
- Statistics about a Logtail configuration for a Logstore. In the configuration, the `file_name` field is set to `logstore_statistics`.

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 globally unique and must be in the following format: <code>##Logtail configuration version##Project name\$Configuration name</code> .	##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-23xeb15ni.1569395.715455,901,00789b
file_dev	The device ID of the log file. Note If the <code>file_name</code> field is set to <code>logstore_statistics</code> , this field is invalid.	123
file_inode	The inode of the log file. Note If the <code>file_name</code> field is set to <code>logstore_statistics</code> , this field is invalid.	124
file_name	The full path of the log file or the value of <code>logstore_statistics</code> .	/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

Field	Description	Example
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 `file_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

Field	Description	Example
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
cpu	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

Field	Description	Example
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

Field	Description	Example
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
plugin_enabled	Indicates whether Logtail plug-ins are enabled.	false
poll_modify_size	The number of monitored log files that are modified.	1
polling_dir_cache	The number of scanned directories.	1
polling_file_cache	The number of scanned files.	1
process_bytes_ps	The size of log data that is processed per second. Unit: bytes.	1000
process_lines_ps	The number of logs that are processed per second.	1000
process_queue_full	The number of processing queues that reach the maximum processing capacity.	1
process_queue_total	The total number of processing queues.	10
process_tps	The number of data processing transactions per second.	0
reader_count	The number of log files that are being processed.	1
region	The region where Logtail resides.	cn-hangzhou,cn-shanghai
register_handler	The number of directories to be monitored.	1
send_bytes_ps	The size of raw log data that is sent per second. Unit: bytes.	11111
send_lines_ps	The number of logs that are sent per second.	1000
send_net_bytes_ps	The volume of network data that is sent per second. Unit: bytes.	1000
send_queue_full	The number of send queues that reach the maximum sending capacity.	1
send_queue_total	The total number of send queues.	12

Field	Description	Example
send_request_concurrency	The maximum number of packets that can be concurrently sent from send queues.	10
send_tps	The number of data sending transactions per second.	0.075
sender_invalid	The number of abnormal send queues.	0
start_time	The start time.	2021-07-20 16:19:22
used_sending_concurrency	The number of packets that are concurrently sent.	0

5.10.4. 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.

5.11. FAQ

5.11.1. Log collection

5.11.1.1. 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 no heartbeat packet is received from a Logtail client?](#).

2. 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.

3. 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 [Set Logtail startup parameters](#).

5.11.1.2. 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.

Context

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:
 - On **Control Panel**, click **Administrative Tools**, and then click **Services**.
 - 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 in Linux](#) or [Install Logtail in Windows](#). 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 following path:

- 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.json* 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:

```
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 in Linux](#) or [Install Logtail in Windows](#).

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, click **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.

5.11.1.3. 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 in Linux](#).

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
command list :
    status all [index]
        get logtail running status
    status active [--logstore | --logfile] index [project] [logstore]
        list all active logstore | logfile. if use --logfile, please add project and logstore. de
fault --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 Set Logtail startup parameters .
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 Set Logtail startup parameters .

active command

- Syntax

 **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 : logstore-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 `--format=` parameter is `--format=line`. 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"
}

```

- Response

Parameter	Description	Unit

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 <code>##1.0## + project + \$ + config</code> .	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

Parameter	Description	Unit
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 Set Logtail startup parameters .
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.

Status	Description	Troubleshooting
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 Set Logtail startup parameters .

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.
- The value of the `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$same
file_path : /disk2/test/normal/access.log
file_dev : 64800
file_inode : 22544456
file_size_bytes : 17154060
file_offset_bytes : 17154060
read_bytes : 65033430
parse_success_lines : 230615
parse_fail_lines : 0
last_read_time : 1503977170
read_count : 667
avg_delay_bytes : 0
/etc/init.d/ilogtaild status logfile --format=json 1 sls-zc-test release-test-same /disk2/test/normal/access.log
{
  "avg_delay_bytes" : 0,
  "config" : "##1.0##sls-zc-test$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_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"
}

```

- 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

Parameter	Description	Unit
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 <code>##1.0## + project + \$ + config</code> .	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

	begin_time		status	read	parse_success	parse_fail	last_read_time	read
_count	avg_delay	device	inode	file_size	read_offset			
17-08-29 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 11:16:11			ok	62.02MB	230615	0	17-08-29 11:26:10	
667	0B	64800	22544456	16.36MB	16.36MB			
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

	begin_time		status	read	parse_success	parse_fail	last_read_time	read
_count	avg_delay	send_queue	network_error	quota_error	discard_error	send_success	send_block	
k	send_valid	max_unsend	min_unsend	max_send_success				
17-08-29 11:16:11			ok	62.02MB	230615	0	17-08-29 11:26:10	
667	0B	0	0	0	0	300	false	
true	70-01-01 08:00:00	70-01-01 08:00:00	17-08-29 11:26:08					
17-08-29 11:06:11			ok	62.12MB	231000	0	17-08-29 11:16:11	
687	0B	0	0	0	0	303	false	
true	70-01-01 08:00:00	70-01-01 08:00:00	17-08-29 11:16:10					
17-08-29 10:56:11			ok	62.02MB	230615	0	17-08-29 11:06:10	
687	0B	0	0	0	0	302	false	
true	70-01-01 08:00:00	70-01-01 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	0	302	false	
true	70-01-01 08:00:00	70-01-01 08:00:00	17-08-29 10:56:10					

- Response

- The collection status of the Logstore or log file in each query time window is listed in a line.
- 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 <code>-h</code> 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 <code>-h</code> command to obtain help information.
1	The collection status in the specified query time window cannot be queried.	query fail, error: \$ (error) . For more information, see errno .	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, may be 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 `logstore` 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 `read_offset_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 [Set Logtail startup parameters](#).
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 [Set Logtail startup parameters](#).
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.

5.11.1.4. 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.

Context

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.

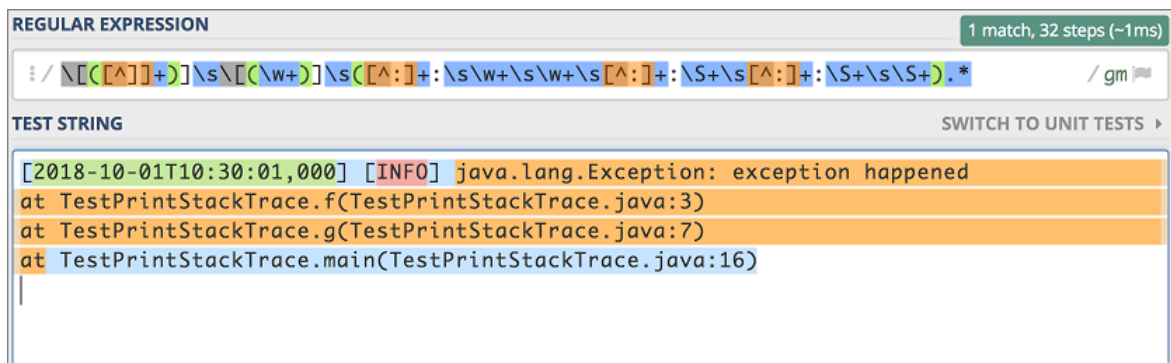
Procedure

1. Log on to the Log Service console.
2. In the Projects section, click the project that you want to manage.
3. In the Logstores list, 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.

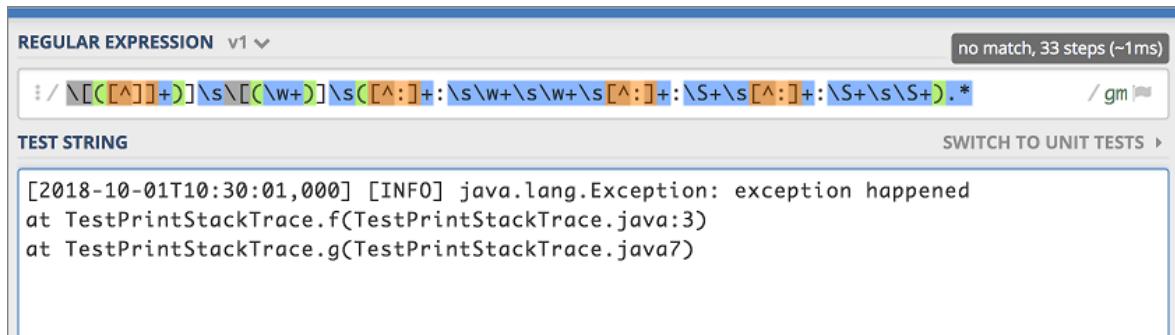
8. In the **TEST STRING** field, paste a sample log entry.

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.



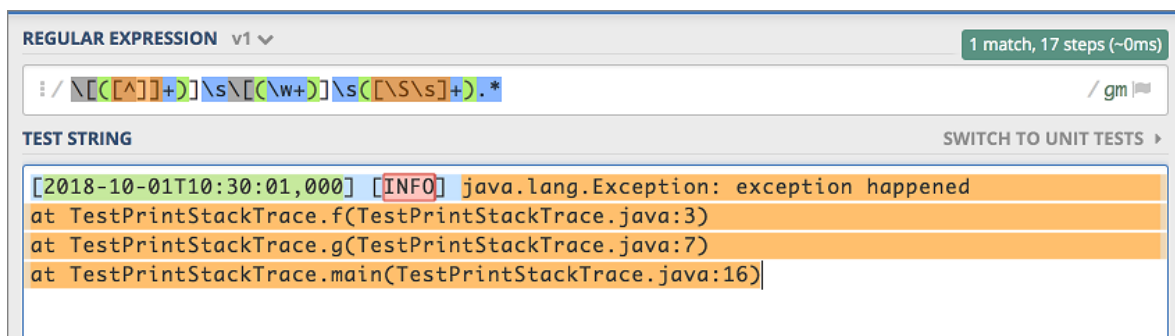
9. Check whether the regular expression can match fields in the sample log entry that contains two colons (::).

The following figure shows that the regular expression fails to match fields in the sample log entry.

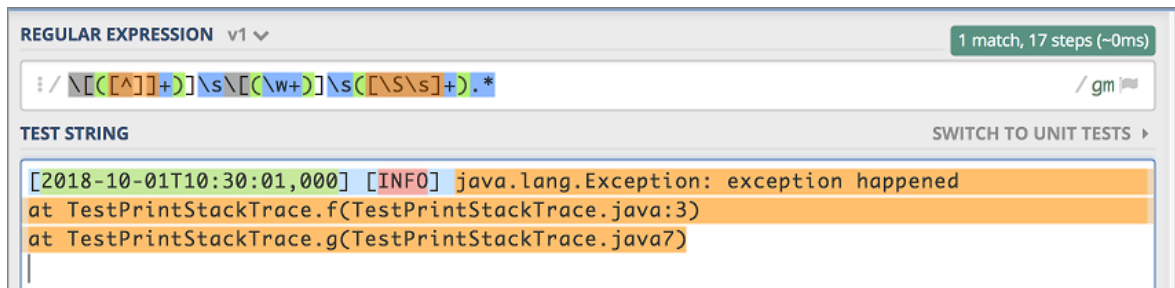


10. Replace the last subexpression in the regular expression with `[\S\s]+`, and check whether the regular expression can match fields in the sample log entries as expected.

The following figure shows how the modified regular expression matches the substring that follows the word.



The following figure shows how the modified regular expression matches the sample log entry that contains two colons (::).



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.

5.11.1.5. 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.com to debug the regular expression. This way, you can troubleshoot the issue and optimize the regular expression in an efficient manner.

5.11.1.6. 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.

 **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.

5.11.1.7. 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
RFC3339     2006-01-02T15:04:05Z07:00
%Y-%m-%dT%H:%M:%S
RFC3339Nano 2006-01-02T15:04:05.999999999Z07:00
%Y-%m-%dT%H:%M:%S

```

5.11.1.8. How do I configure non-printable characters in a sample log?

This topic describes how to configure non-printable characters in a sample log.

Context

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: `0xHexadecimal ASCII code of the non-printable character`. For example, a sample log is `123456780`. You can specify `0x01` as 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 DSV formatted logs](#).

* Log Sample: 12345""67890

The sample log entry is different from the original entry. [Modify Sample Log Entry](#)

7. Delete the double quotation marks (") between the digits 5 and 6.

A non-printable character is configured in a sample log.

* Log Sample: 1234567890

The sample log entry is different from the original entry. [Modify Sample Log Entry](#)

5.11.1.9. 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.
 - 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, click the **Machine Groups** icon.
 - 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.
 - 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	VERSION
cn-hangzhou.i-bp17enxc2us3624wexh2	Ready	<none>	238d	v1.10.4
cn-hangzhou.i-bp1ad2b02jtd1shi2ut	Ready	<none>	220d	v1.10.4

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.
 - The heartbeat status of all servers in the machine group is **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_machine_group_user_defined_id}` parameters are valid. For more information, see the Parameters section in [Collect standard Docker logs](#).
- 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 re-install 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
```

 **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$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500f8e2bdb95d13b1e110172ef57fe8
40c82155/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44bea
b6e69718/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640b1e
16c22dbe/merged: must be superuser to unmount
.....
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
ilogtail 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" : "0EBB2B0E-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"
}
```

5.11.1.10. 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

1. 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.
2. 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@iZbp14up92567375kxjqz ~]# docker ps | grep orders
2ba4ebdaf503      43e27feaa78a      "/usr/local/bin/java..." 2 months ago      Up 2 months
k8s_orders_orders-7895d5f946-s6xxj_victor-center_2348cd71-6a91-4b5f-af26-73fc03a9c571_0
registry-vpc.cn-hangzhou.aliyuncs.com/acs/pause-amd64:3.0      "/pause"      2 months ago      Up 2 months
k8s_POD_orders-7895d5f946-s6xxj_victor-center_2348cd71-6a91-4b5f-af26-73fc03a9c571_0
```

3. 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.

```
docker inspect 2ba4ebdaf503
```

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-s6xxj_2348cd71-6a91-4b5f-af26-73fc03a9c571/orders/0.log",
    "io.kubernetes.container.name": "orders",
    "io.kubernetes.docker.type": "container",
    "io.kubernetes.pod.name": "orders-7895d5f946-s6xxj",
    "io.kubernetes.pod.namespace": "victor-center",
    "io.kubernetes.pod.uid": "2348cd71-6a91-4b5f-af26-73fc03a9c571",
    "io.kubernetes.sandbox.id": "0778af11-1111-1111-1111-228983eafcc0e9a274c3eef83e785f298568",
    "msd_java_build_commit": "05998875b01011111111111111111111113e15b2e7",
    "msd_java_build_date": "2017-11-21T12:52:16+0000",
    "msd_java_build_version": "0.0.2-SNAPSHOT"
  },
  "NetworkSettings": {
    "Bridge": "",
    "SandboxID": "",
    "HairpinMode": false,
```

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.
2. 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@iZbp14up92567375kxjqz ~]# docker ps | grep orders
2ba4ebdaf503      43e27feaa78a      "/usr/local/bin/java..." 2 months ago      Up 2 months
k8s_orders_orders-7895d5f946-s6xxj_victor-center_2348cd71-6a91-4b5f-af26-73fc03a9c571_0
registry-vpc.cn-hangzhou.aliyuncs.com/acs/pause-amd64:3.0      "/pause"      2 months ago      Up 2 months
k8s_POD_orders-7895d5f946-s6xxj_victor-center_2348cd71-6a91-4b5f-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=172.21.1.225
PAYMENT_PORT_80_TCP=tcp://172.21.1.24:5:80
CATALOGUE_DB_SERVICE_HOST=172.21.1.96
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=172.21.1.225
USER_PORT_80_TCP_PROTO=tcp
MONGO_PORT_27017_TCP_ADDR=172.21.1.199
PAYMENT_PORT=tcp://172.21.1.24:5:80
CARTS_PORT_80_TCP=tcp://172.21.1.5:30:80
INTEGRAL_PORT=tcp://172.21.1.13:80
CATALOGUE_PORT_80_TCP=tcp://172.21.1.113:80
USER_PORT_80_TCP=tcp://172.21.1.101:80
KUBERNETES_PORT=tcp://172.21.1.443
CARTS_PORT=tcp://172.21.1.5:30:80
TEST_PORT_27017_TCP_ADDR=172.21.1.60
INTEGRAL_PORT_80_TCP_ADDR=172.21.1.12.53
CATALOGUE_SERVICE_HOST=172.21.1.133
SESSION_DB_PORT=tcp://172.21.1.249:6379
INTEGRAL_SERVICE_PORT_INTEGRAL=80
RABBITMQ_PORT_5672_TCP_PORT=5672
RABBITMQ_PORT_5672_TCP_ADDR=172.21.1.52
ORDERS_SERVICE_PORT=80
TEST_PORT_27017_TCP_PORT=27017
CATALOGUE_DB_PORT_3306_TCP=tcp://172.21.1.6:3306
INTEGRAL_DB_SERVICE_PORT=3306
INTEGRAL_SERVICE_PORT=80
CARTS_SERVICE_HOST=172.21.1.30
CARTS_SERVICE_PORT=80
CATALOGUE_PORT_80_TCP_PROTO=tcp
SESSION_DB_PORT_6379_TCP_PORT=6379

```

5.11.2. Log search and analysis

5.11.2.1. 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?

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]:&nbsp;device_id:&nbsp;BTAddr&nbsp;:&nbsp;B6:xF:xx:65:xx:A1&nbsp;IMEI&nbsp;:&nbsp;35847xx22
xx81x9&nbsp;WifiAddr&nbsp;:&nbsp;4c:xx:0e:xx:4e:xx&nbsp;|&nbsp;user_id:&nbsp;bb07263xxd2axx43xx9exxea2
6e39e5f&nbsp;POS&nbsp;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.

5.11.2.2. 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 no heartbeat packet is received 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 `,;=()[]{}?@&<>/:/'` 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 [Query 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.

5.11.2.3. 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.

5.11.2.4. 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 Enable the indexing feature and configure indexes for a Logstore .
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.

Error message	Cause	Solution
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 <pre>select key1, avg(latency) group by key2</pre> statement in the GROUP BY clause.	You must specify the same field that you specified in the SELECT statement in the GROUP BY clause. Example: <pre>* select key1,avg(latency) group by key1,key2</pre>
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.	Invalid query statement: <pre>select ip,count(*) group by ip</pre> . Valid query statement: <pre>* 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 Enable the indexing feature and configure indexes for a Logstore .
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 <code>GROUP BY</code> clause to remove duplicates.	Optimize the <code>GROUP BY</code> clause. Reduce the number of keys that is specified in the <code>GROUP BY</code> clause.
ErrorType:ColumnNotExists.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 Enable the indexing feature and configure indexes for a Logstore . 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 message	Cause	Solution
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'}	The syntax of the query statement is valid.	Modify the SQL statement as prompted and execute the SQL statement.
Duplicate keys (XXX) are not allowed	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 <code>Msg: *xxx</code> 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: <pre>Msg: * SELECT Msg WHERE Msg LIKE '%xxx'</pre>
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.

5.11.2.5. 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.

5.11.2.6. 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.

 **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.

5.11.3. Alarm

5.11.3.1. 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  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 dashboard page, choose **Alerts > Create**.
6. In the **Create Alert** panel, set the parameters. The following figure shows the parameter settings.

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: `$1.count > 5`
- Content: `${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 [DingTalk chatbot webhooks](#). If the issue persists or other error messages are returned, submit a ticket.

5.11.4. What do I do if the Forbidden.SLS::ListProject error occurs when I log on to the Apsara Uni-manager Management Console?

Error description

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.

Cause

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.

Solution

Procedure

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.