

Alibaba Cloud

Apsara Stack Enterprise

Apsara Stack Enterprise Product Introduction

Product Version: 2105, Internal: V3.14.0

Document Version: 20210818

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.Introduction to Alibaba Cloud Apsara Stack	21
1.1. What is Apsara Stack?	21
1.2. The reasons to choose Apsara Stack	22
1.2.1. Hyper-scale distributed cloud operating system	22
1.2.2. Apsara Infrastructure Management Framework	23
1.2.3. Centralized cross-region management of multiple data	24
1.2.4. Highly reliable disaster recovery solutions	25
1.2.5. Centralized operations management and automated O&... ..	33
1.2.6. Rolling upgrade	33
1.2.7. OpenAPI	33
1.3. Product architecture	33
1.3.1. Types of private cloud architectures	34
1.3.2. System architecture	34
1.3.3. Network architecture	36
1.3.3.1. Network architecture overview	36
1.3.3.2. Business service section	37
1.3.3.3. Integrated access section	38
1.3.3.4. VPC leased line access	40
1.3.4. Security architecture	41
1.3.5. Base modules	41
1.4. Product panorama	43
1.5. Scenarios	43
1.6. Compliance security solution	44
1.6.1. Overview	45
1.6.2. Interpretation of key points	45
1.6.3. Cloud-based classified protection compliance	46

1.6.4. Classified protection implementation process	48
1.6.5. Security compliance architecture	49
1.6.6. Solution benefits	50
2.Elastic Compute Service (ECS)	52
2.1. What is ECS?	52
2.2. Benefits	52
2.3. Architecture	54
2.4. Features	55
2.4.1. Instances	55
2.4.1.1. Overview	55
2.4.1.2. Instance families	56
2.4.1.3. Instance types	75
2.4.1.4. UserData	104
2.4.1.5. Instance lifecycle	104
2.4.1.6. EBM Instances	106
2.4.1.7. Super Computing Clusters	107
2.4.2. Block storage	109
2.4.2.1. Overview	109
2.4.2.2. Elastic block storage	110
2.4.2.2.1. Overview	110
2.4.2.2.2. Cloud disks	110
2.4.2.2.3. Shared Block Storage	111
2.4.2.2.4. Triplicate storage	112
2.4.2.2.5. Erasure coding	113
2.4.2.3. ECS disk encryption	114
2.4.2.4. Local storage	114
2.4.3. Images	115
2.4.4. Snapshots	116

2.4.4.1. Overview	116
2.4.4.2. Mechanisms	116
2.4.4.3. Specifications of ECS Snapshot 2.0	117
2.4.5. Deployment sets	118
2.4.6. Network and security	120
2.4.6.1. IP addresses of ECS instances of VPC type	120
2.4.6.2. Elastic network interfaces	121
2.4.6.3. Internal network	122
2.4.6.4. Security group rules	122
2.5. Scenarios	123
2.6. Terms	123
3.Container Service for Kubernetes	125
3.1. What is Container Service?	125
3.2. Benefits	125
3.3. Architecture	127
3.4. Features	128
3.5. Scenarios	129
3.6. Limits	133
3.7. Terms	134
4.Auto Scaling (ESS)	137
4.1. What is Auto Scaling?	137
4.2. Benefits	138
4.3. Architecture	139
4.4. Features	140
4.5. Scenarios	141
4.6. Limits	141
4.7. Terms	142
5.Resource Orchestration Service (ROS)	143

5.1. What is ROS?	143
5.2. Benefits	143
5.3. Architecture	144
5.4. Features	145
5.5. Scenarios	146
5.6. Limits	147
5.7. Terms	147
6.Object Storage Service (OSS)	148
6.1. What is OSS?	148
6.2. Benefits	148
6.3. Architecture	149
6.4. Terms	150
6.5. Features	152
6.6. Scenarios	153
6.7. Limits	154
7.Tablestore	155
7.1. What is Tablestore?	155
7.2. Benefits	155
7.3. Architecture	157
7.4. Scenarios	158
7.5. Limits	164
7.6. Terms	165
7.7. Features	167
7.7.1. Features	167
7.7.2. Global secondary index	168
7.7.2.1. Features	168
7.7.2.2. Usage notes	169
7.7.2.3. Scenarios	170

8.ApsaraDB RDS	180
8.1. What is ApsaraDB RDS?	180
8.2. Benefits	181
8.2.1. Ease of use	181
8.2.2. High performance	181
8.2.3. High security	181
8.2.4. High reliability	182
8.3. Architecture	183
8.4. Features	183
8.4.1. Data link service	183
8.4.2. High-availability service	184
8.4.3. Backup and recovery service	185
8.4.4. Monitoring service	186
8.4.5. Scheduling service	186
8.4.6. Migration service	187
8.5. Scenarios	188
8.5.1. Diversified data storage	188
8.5.2. Read/write splitting	189
8.5.3. Big data analysis	190
8.6. Limits	191
8.6.1. Limits on ApsaraDB RDS for MySQL	191
8.6.2. Limits on ApsaraDB RDS for SQL Server	192
8.6.3. Limits on ApsaraDB RDS for PostgreSQL	193
8.6.4. Limits on PolarDB	193
8.7. Terms	194
8.8. Instance types	195
9.Cloud Native Distributed Database PolarDB-X	210
9.1. What is PolarDB-X?	210

9.2. Benefits	210
9.3. Architecture	212
9.4. Features	214
9.4.1. Scalability	214
9.4.2. Distributed transactions	216
9.4.3. Smooth scale-out	217
9.4.4. Read/write splitting	218
9.4.5. Global secondary index	219
9.5. Scenarios	220
9.6. Limits	221
9.7. Terms	221
10. AnalyticDB for PostgreSQL	224
10.1. What is AnalyticDB for PostgreSQL?	224
10.2. Benefits	225
10.3. Architecture	226
10.4. Specifications	227
10.5. Features	228
10.6. Scenarios	232
10.7. Introduction to AnalyticDB for PostgreSQL V6.0	234
11. KVStore for Redis	236
11.1. What is KVStore for Redis?	236
11.2. Enhanced Edition and supported commands	236
11.2.1. Performance-enhanced instances of KVStore for Redis ...	236
11.2.2. CAS and CAD commands	240
11.2.3. TairString commands	242
11.2.4. TairHash commands	251
11.2.5. TairGIS commands	273
11.2.6. TairBloom commands	281

11.2.7. TairDoc commands	287
11.3. Benefits	299
11.4. Architecture	300
11.5. Features	301
11.6. Scenarios	303
11.7. Limits	304
11.8. Terms	304
11.9. Instance specifications	305
12.ApsaraDB for MongoDB	311
12.1. What is ApsaraDB for MongoDB?	311
12.2. Benefits	311
12.3. System architecture	312
12.3.1. ApsaraDB for MongoDB system architecture	312
12.3.2. Architecture of replica set instances	313
12.3.3. Architecture of sharded cluster instances	314
12.4. Features	315
12.5. Scenarios	316
12.6. Limits	317
12.7. Terms	317
12.8. Instance types	318
13.Data Management (DMS)	322
13.1. What is DMS?	322
13.2. Benefits	322
13.3. Architecture	323
13.4. Features	323
13.4.1. Workbench	323
13.4.2. Top search box	324
13.4.3. Permission application	324

13.4.4. Data Plans	324
13.4.4.1. Data Changes	324
13.4.4.2. Change schemas without locking tables	325
13.4.4.3. Data import	325
13.4.4.4. Data export	326
13.4.4.5. Test data generation	326
13.4.4.6. Database cloning	326
13.4.5. Data factory	327
13.4.5.1. Task orchestration	327
13.4.5.2. Data warehouse development	327
13.4.5.3. Data services	328
13.4.6. Schemas	328
13.4.6.1. Synchronization between tables and databases	328
13.4.6.2. Schema design	329
13.4.7. SQL reviews	329
13.4.8. SQLConsole	329
13.4.9. Cross-instance queries	330
13.4.10. System management	330
13.4.10.1. Instance management	330
13.4.10.2. User management	331
13.4.10.3. Task management	331
13.4.10.4. Security rules	331
13.4.10.5. Approval processes	332
13.4.10.6. Operational logs	332
13.4.10.7. IP whitelisting	332
13.4.10.8. Sensitive data management	333
13.4.10.9. Row-level sensitive data management	333
13.4.10.10. Configuration management	333

13.5. Terms	334
14.Server Load Balancer (SLB)	347
14.1. What is SLB?	347
14.2. High availability	348
14.3. Architecture	349
14.4. Features	352
14.5. Scenarios	354
14.6. Limits	355
14.7. Terms	355
15.Virtual Private Cloud (VPC)	357
15.1. What is a VPC?	357
15.2. Benefits	358
15.3. Architecture	358
15.4. Features	360
15.5. Scenarios	361
15.6. Terms	363
15.7. Limits	363
16.NAT Gateway	366
16.1. What is NAT Gateway?	366
16.2. Features	366
16.3. Benefits	367
16.4. Scenarios	367
16.5. Terms	369
16.6. Limits	369
17.VPN Gateway	371
17.1. What is VPN Gateway?	371
17.2. Scenarios	372
17.3. Limits	374

18.Elastic IP Address	376
18.1. What is Elastic IP Address?	376
18.2. Limits	376
19.Log Service	379
19.1. What is Log Service?	379
19.2. Benefits	379
19.3. Architecture	380
19.4. Features	381
19.4.1. Core features	381
19.4.2. Other features	382
19.4.2.1. Logs	382
19.4.2.2. Project	384
19.4.2.3. Logstore	385
19.4.2.4. Shard	385
19.4.2.5. Log topic	387
19.5. Scenarios	387
19.6. Limits	388
19.7. Terms	389
20.Apsara Stack Security	391
20.1. What is Apsara Stack Security	391
20.2. Benefits	392
20.3. Architecture	393
20.4. Features	395
20.4.1. Apsara Stack Security Standard Edition	395
20.4.2. On-premises security operations services	400
20.4.3. Optional security services	401
20.5. Restrictions	403
20.6. Terms	403

21.Key Management Service (KMS)	404
21.1. What is KMS?	404
21.2. Benefits	405
21.3. Scenarios	406
21.4. Terms	406
22.Apsara Stack DNS	408
22.1. What is Apsara Stack DNS?	408
22.2. Edition comparison	408
22.3. Benefits	410
22.4. Architecture	411
22.5. Features	413
22.6. Scenarios	415
22.7. Limits	417
22.8. Terms	419
23.API Gateway	420
23.1. What is API Gateway?	420
23.2. Features	420
23.3. Benefits	421
23.4. Terms	422
24.MaxCompute	424
24.1. What is MaxCompute?	424
24.2. Integration with other Alibaba Cloud services	424
24.3. Benefits	427
24.4. Architecture	429
24.5. Features	432
24.5.1. Tunnel	432
24.5.1.1. Terms	433
24.5.1.2. Tunnel features	433

24.5.1.3. Data upload and download through Tunnel	433
24.5.2. SQL	434
24.5.2.1. Terms	434
24.5.2.2. SQL characteristics	434
24.5.2.3. Comparison with open source products	434
24.5.3. MapReduce	435
24.5.3.1. Terms	435
24.5.3.2. MapReduce characteristics	435
24.5.3.3. MaxCompute MapReduce process	436
24.5.3.4. Hadoop MapReduce VS MaxCompute MapReduce	436
24.5.4. Graph	437
24.5.4.1. Terms	437
24.5.4.2. Graph characteristics	437
24.5.4.3. Graph relational network models	437
24.5.5. Unstructured data processing in integrated computing.. ..	438
24.5.6. Unstructured data processing in MaxCompute	439
24.5.7. Enhanced features	439
24.5.7.1. Spark on MaxCompute	439
24.5.7.1.1. Terms	439
24.5.7.1.2. Features of Spark on MaxCompute	439
24.5.7.1.3. Spark features	440
24.5.7.1.4. Spark architecture	440
24.5.7.1.5. Benefits of Spark on MaxCompute	441
24.5.7.2. Elasticsearch on MaxCompute	442
24.5.7.2.1. Overview	442
24.5.7.2.2. Features of Elasticsearch on MaxCompute	442
24.5.7.2.3. Elasticsearch features	443
24.5.7.2.4. Elasticsearch architecture	443

24.5.7.2.5. Benefits	444
24.5.8. Multi-region deployment	445
24.5.8.1. MaxCompute multi-region deployment	445
24.6. Scenarios	446
24.6.1. Scenario 1: Migrate data to the cloud cost-effectively	446
24.6.2. Scenario 2: Improve development efficiency and reduc... ..	447
24.6.3. Scenario 3: Use mass data to achieve precision marke... ..	447
24.6.4. Scenario 4: Achieve precision marketing with big data	448
24.7. Limits	448
24.8. Terms	448
24.9. Storage performance	451
25.DataWorks	455
25.1. What is DataWorks?	455
25.2. Benefits	455
25.3. Architecture	456
25.3.1. Service architecture	456
25.3.2. System architecture	457
25.3.3. Security architecture	457
25.3.4. Multitenancy	457
25.4. Services	457
25.4.1. Data Integration	457
25.4.2. DataStudio	459
25.4.2.1. Overview	459
25.4.2.2. Workflows	459
25.4.2.3. Solutions	460
25.4.2.4. Code editor	460
25.4.2.5. Code repository and team collaboration	461
25.4.3. Administration	462

25.4.3.1. Overview	462
25.4.3.2. Overview page	462
25.4.3.3. Node O&M pages	462
25.4.3.4. Intelligent Monitor service	462
25.4.3.5. Engine O&M	463
25.4.4. DataAnalysis	463
25.4.5. Data Map	464
25.4.6. Security Center	464
25.4.7. DataService Studio	464
25.4.8. Migration Assistant	465
25.4.9. Workspace Management	466
25.4.10. Data Asset Management	467
25.4.11. Data Protection	467
25.4.11.1. Overview	467
25.4.11.2. Terms	468
25.4.11.3. Management	468
25.4.11.4. Data recognition	469
25.4.11.5. Data Activities	469
25.4.11.6. Data masking	469
25.4.11.7. Levels	469
25.4.11.8. Manual check	469
25.4.11.9. Data risks	470
25.4.11.10. Risk Rules	470
25.4.11.11. Data Auditing	470
25.5. Scenarios	470
25.5.1. Cloud-based data warehouse	470
25.5.2. Business intelligence	470
25.5.3. Data-driven management	471

25.6. Limits	471
25.7. Terms	471
26. Realtime Compute	476
26.1. What is Realtime Compute?	476
26.2. End-to-end real-time computing	477
26.3. Differences between real-time computing and batch comp..	477
26.3.1. Overview	477
26.3.2. Batch computing	478
26.3.3. Real-time computing	478
26.3.4. Comparison between real-time computing and batch ...	479
26.4. Benefits	480
26.5. Product architecture	482
26.5.1. Workflow	482
26.5.2. Business architecture	484
26.5.3. Technical architecture	485
26.6. Features	486
26.7. Product positioning	488
26.8. Scenarios	488
26.8.1. Overview	488
26.8.2. Management of e-commerce activities	489
26.8.3. Multidimensional analysis of data from IoT sensors	489
26.8.4. Big screen service for the Tmall Double 11 Shopping ...	494
26.8.5. Mobile data analysis	495
26.9. Restrictions	496
26.10. Basic concepts	496
27. Machine Learning Platform for AI	498
27.1. What is Machine Learning Platform for AI?	498
27.2. Benefits	499

27.3. Architectures	500
27.4. Features	502
27.4.1. Visualized modeling	502
27.4.2. All-in-one experience	503
27.4.3. Multiple templates on the homepage	504
27.4.4. Data visualization	505
27.4.5. Model management	505
27.4.6. Algorithm components	506
27.4.7. Data labeling	518
27.4.8. DSW	519
27.5. Scenarios	522
27.6. Limits	523
27.7. Terms	524
28.DataHub	526
28.1. What is DataHub?	526
28.2. Benefits	526
28.3. Architecture	527
28.4. Features	528
28.4.1. Data queue	528
28.4.2. Checkpoint-based data restoration	528
28.4.3. Data synchronization	528
28.4.4. Scalability	529
28.5. Scenarios	529
28.5.1. Overview	529
28.5.2. Data uploading	529
28.5.3. Data collection	530
28.5.4. Realtime Compute	530
28.5.5. Data utilization	531

28.5.6. Data archiving	531
28.6. Limits	531
28.7. Terms	532
29.Apsara Big Data Manager (ABM)	535
29.1. What is Apsara Big Data Manager?	535
29.2. Benefits	535
29.3. Architecture	535
29.3.1. O&M Architecture	535
29.4. Features	537
29.4.1. Dashboard	537
29.4.2. Repository	540
29.4.3. O&M	542
29.4.4. Management	546
29.5. Scenarios	547
29.6. Limits	547
29.7. Concepts	548
30.Apsara Uni-manager	550
30.1. What is Apsara Uni-manager?	550
30.2. Benefits	550
30.3. Architecture	552
30.4. Features	553
30.5. Scenarios	554

1. Introduction to Alibaba Cloud Apsara Stack

1.1. What is Apsara Stack?

Apsara Stack is an open, unified, and trusted cloud platform tailored for enterprise customers. It is developed based on the same distributed architecture as Alibaba Cloud public cloud. Enterprises customers can deploy public cloud products and services in on-premises environments, expand capabilities to the public cloud with ease, and enjoy hybrid cloud services anytime and anywhere.

Private clouds

A private cloud is a cloud computing system that is deployed on premises for government and enterprise customers by cloud computing service providers. Cloud infrastructure, software, and hardware resources are deployed in the private cloud behind a firewall to allow internal users of enterprises to share resources within their data centers. The private cloud can be managed by the enterprise itself or by a third party, and located within or outside the enterprise. Private clouds provide better privacy and exclusivity than public clouds.

Private clouds are divided into the following types based on enterprise scale or business requirements:

- Multi-tenant comprehensive private clouds for industries and large groups: an end-to-end cloud system created in a top-down manner. The system is designed to drive hyper-scale digital applications and meet IT requirements such as the continuous integration and development of DevOps applications and the operation support of production environments.
- Single-tenant basic private clouds for small and medium-sized enterprises and scenarios: a cloud system that can perform local computing tasks and host technical systems such as large-scale Software as a Service (SaaS) applications, industrial clouds, and large group clouds.

Apsara Stack

More and more enterprises migrate their IT infrastructure to the cloud, and they must consider construction requirements such as security compliance, reuse of existing data centers, and the benefits of a collocated data center. Some enterprises may prefer to use their own data centers but want to deliver a service experience that relies on large-scale cloud computing.

Apsara Stack is an extension of Alibaba Cloud public cloud that brings public cloud technology to private clouds. Apsara Stack delivers complete, customizable software solutions based on Alibaba Cloud and allows enterprises to experience the hyper-scale cloud computing and big data products of Alibaba Cloud public cloud within their own data centers. Apsara Stack also provides enterprises with a consistent hybrid cloud experience where you can obtain IT resources and ensure business continuity.

Apsara Stack supports local deployment and can be independently run and managed outside Alibaba Cloud.

Benefits

Apsara Stack helps government and enterprise customers digitally transform their businesses and services based on a variety of products, services, and the digitalization practices of Alibaba Group in combination with mature solutions and developed from experience. Apsara Stack provides the following benefits:

- Elastic

Combines all resources into a single supercomputer and flexibly scales resources to minimize costs and maximize performance and stability.

- Agile

Uses Internet and microservice integration to accelerate innovation.

- Ultra-large scale

Allows you to deploy tens of thousands of servers within a single region with multi-tenant support. Apsara Stack enables you to run services across multiple regions and is capable of handling tremendous volumes of businesses.

- Secure and reliable

Adopts a hierarchical security model to offer multi-level integrated security protection. Apsara Stack also offers finance-grade disaster recovery plans, which ensures high system availability and business continuity.

- Online operable

Provides industry cloud services that support accurate metering and billing. Apsara Stack enables enterprises to choose products and services and manage them easily.

Platform features

Apsara Stack is an enterprise-level cloud platform. It has the following features:

- Software-defined platform: masks underlying hardware differences, enables resources to scale up or out, and does not affect the performance of upper-layer applications.
- Production-level reliability and security compliance: ensures the continuity and security of enterprise data.
- Centralized access management: isolates permissions of different roles to facilitate subsequent O&M management.

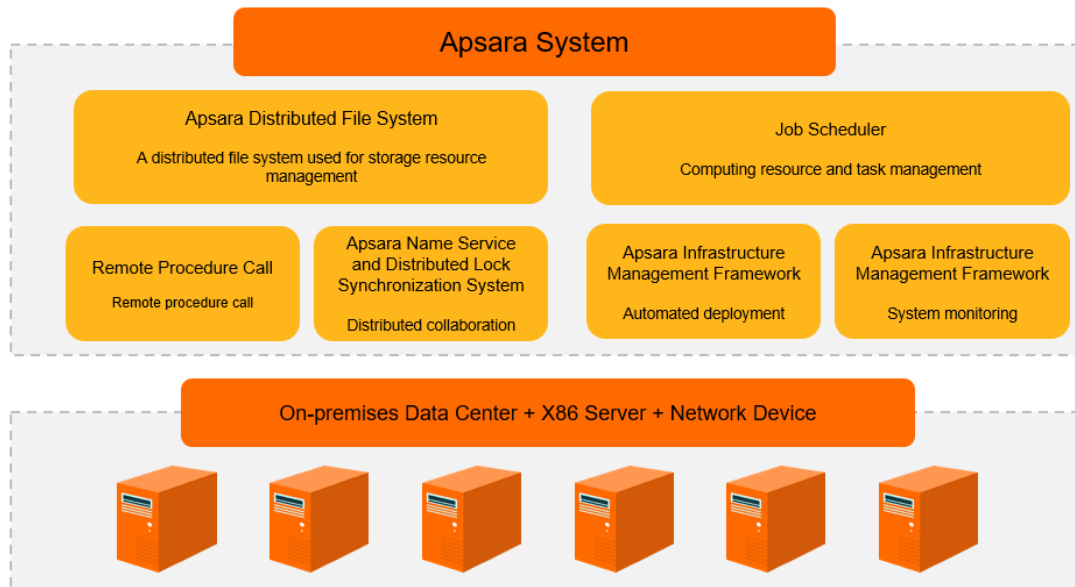
1.2. The reasons to choose Apsara Stack

1.2.1. Hyper-scale distributed cloud operating system

Both Apsara Stack and Alibaba Cloud public cloud are based on the Apsara system. The Apsara system provides underlying services such as storage, computing, and scheduling services for upper-layer services.

The Apsara system is a hyper-scale universal operating system developed by Alibaba Cloud for use both inside and outside China. It connects millions of servers around the world to act as a supercomputer. This provides powerful, universal, and accessible computing capabilities as online public services.

Apsara system kernel architecture



The Apsara system kernel consists of the following modules:

- Underlying services for distributed systems

This module provides coordination, remote procedure call, security management, and resource management services needed in a distributed environment. These services provide support for upper-layer modules such as the distributed file system and task scheduling module.

- Distributed file system

This module provides a reliable and scalable service to store large amounts of data. The distributed file system aggregates the storage capabilities of each node in a cluster and automatically protects against hardware and software faults to provide uninterrupted access to data. This module also supports incremental scaling and automatic data balancing. An API similar to Portable Operating System Interface of UNIX (POSIX) is provided to access user space files. Additionally, the module supports random read/write and append write operations.

- Task scheduling

This module schedules tasks in the cluster system and supports both online services that rely on a quick response speed and offline tasks that require high data processing throughput. The module can automatically detect faults and hot spots in the system. The module ensures stable and reliable service operations by using methods such as error retry and concurrent backup for long-tail operations.

- Cluster monitoring and deployment

This module monitors the status of clusters as well as the running status and performance metrics of upper-layer application services, and generates alerts and records of exception events. In addition, this module provides O&M personnel with deployment and configuration management of the entire Apsara system and upper-layer applications. The module supports both the online elastic scaling of clusters and the online upgrade of application services.

1.2.2. Apsara Infrastructure Management Framework

Apsara Infrastructure Management Framework provides cloud services with underlying support capabilities such as unified deployment, verification, authorization, and control.

Apsara Infrastructure Management Framework includes modules such as deployment framework, resource library, metadatabase, authentication and authorization, API Gateway, Log Service, and control service.

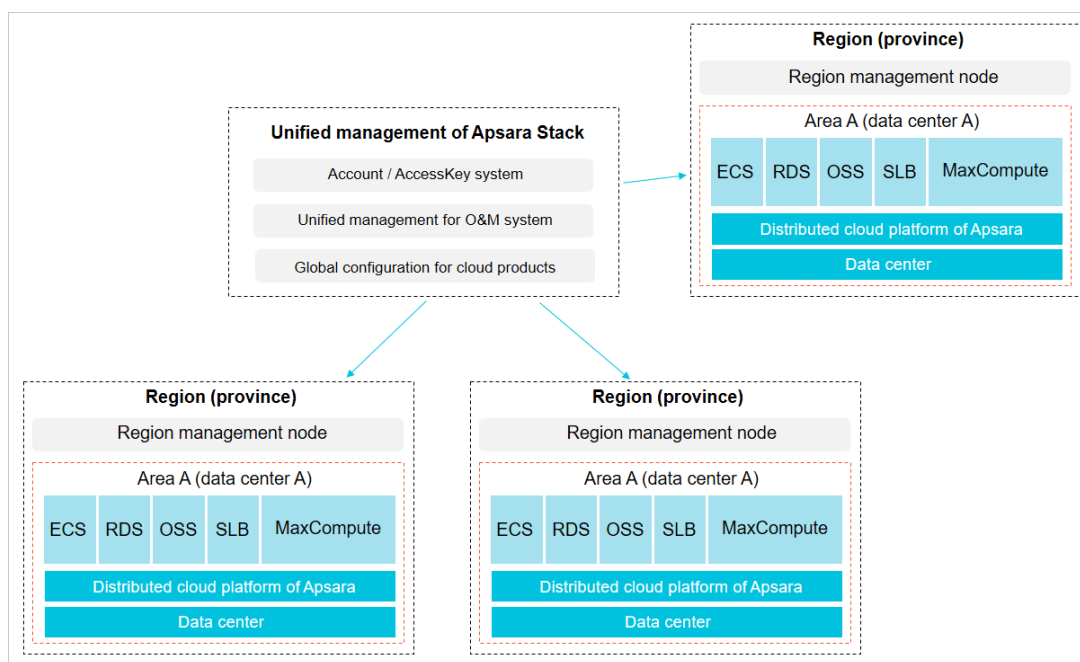
- The deployment framework provides unified deployment of access platforms and manages service dependencies.
- The resource library stores the executable files of all cloud services and their dependent components.
- The authentication and authorization module provides access control capabilities for cloud services and supports multi-tenant isolation.
- API Gateway provides a centralized API management platform for cloud services.
- Log Service provides log storage, retrieval, and access for cloud services.
- The control service module monitors the basic health status of each cloud service and supports the Apsara Stack O&M system.

1.2.3. Centralized cross-region management of multiple data centers

Apsara Stack implements centralized management for O&M, operations, and metering in each region.

Express Connect supports cross-region data access and sharing between two VPCs. This transforms cloud computing into a basic service like water, electricity, and coal provided in a country and brings benefits to everyone. The Apsara Stack management system that converges resource pools can centralize and monitor the computing, storage, and network resources as well as their usage of multiple data centers. This system provides centralized capabilities, including resource management, resource deployment, O&M management, service management, and self-services.

Centralized cross-region management of multiple data centers



1.2.4. Highly reliable disaster recovery solutions

Apsara Stack provides a variety of solutions for disaster recovery (DR), such as geo-disaster recovery, zone-disaster recovery, Hybrid Backup Recovery (HBR), hybrid networking of multi-region deployment and DR, and DR within three data centers across two regions.

A DR system includes two or more systems that provide the same features in distant locations. These systems mutually monitor health status and switch features. If one system stops due to an unexpected incident (such as a fire, flood, earthquake, or vandalism), services can be failed over to a system in a different location to ensure that your business continues normally.

Apsara Stack DR solutions are designed and developed based on the cloud computing capabilities of Alibaba Cloud. These solutions comply with common international DR standards. When the network conditions meet the design requirements, the Apsara Stack platform implements the active-active mode on the network access and user application layers and the active-standby mode on the data persistence layer.

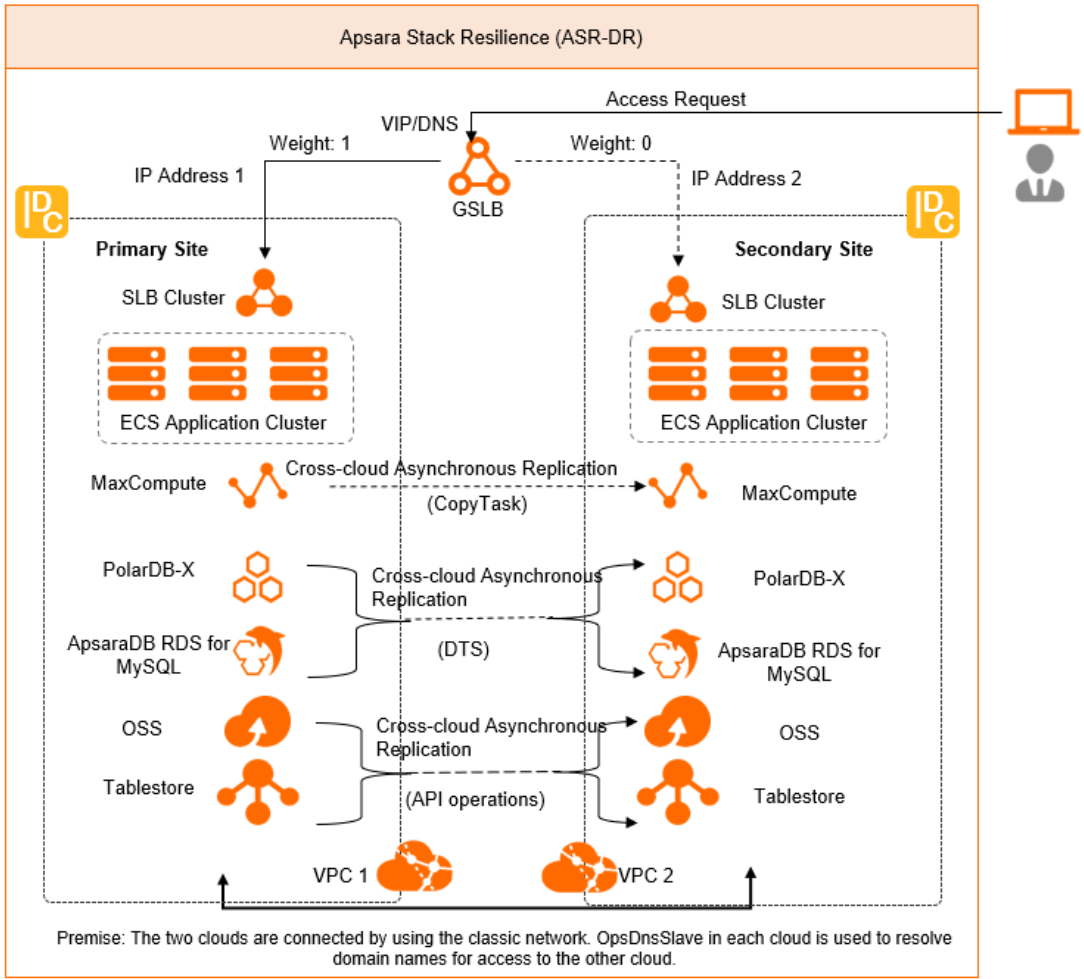
HBR offers a solution that backs up the data of a production center by using a distant backup and disaster recovery (BDR) center. If data in the production center is partially or completely destroyed due to an unexpected incident (such as a fire, flood, earthquake, or vandalism), the HBR system can use the backup data to recover businesses. This ensures business continuity and data security.

Geo-disaster recovery

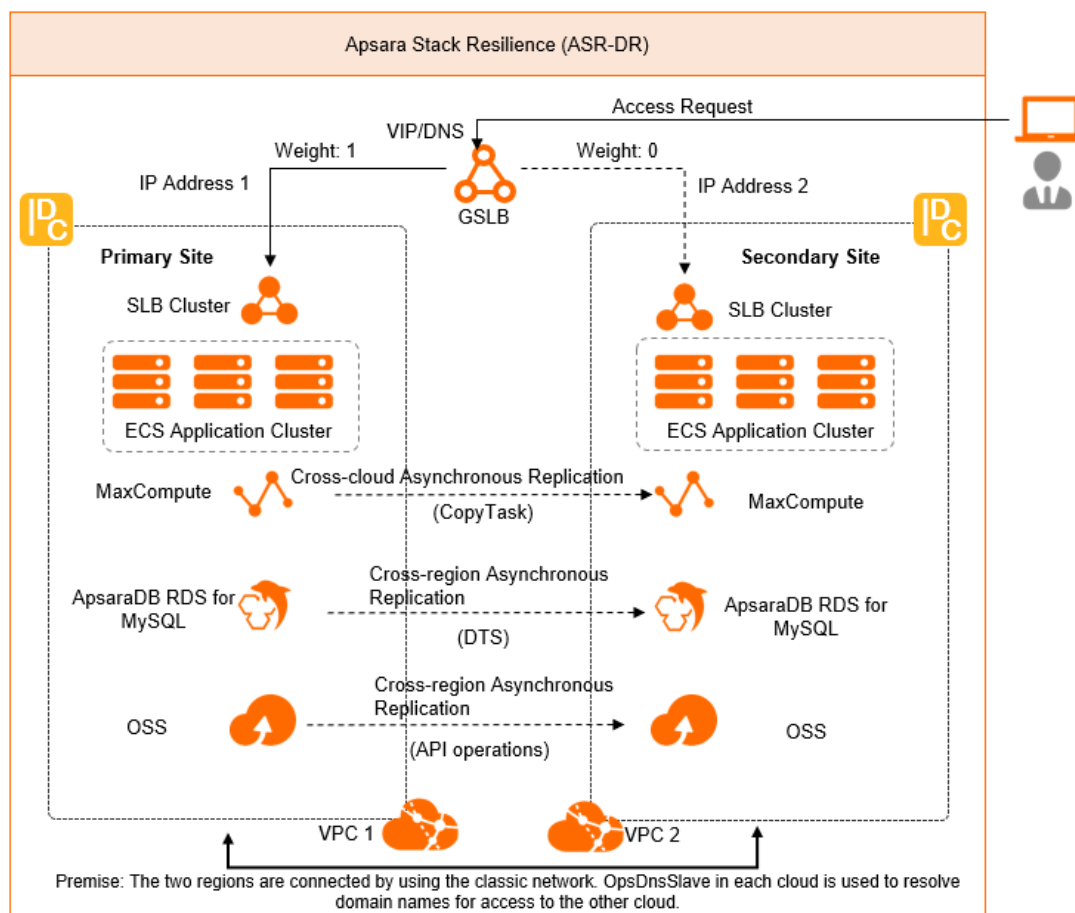
Geo-disaster recovery solutions use the active-standby mode, in which the resources of both primary and secondary data centers are available to users. Protected resources such as ApsaraDB RDS instances and Object Storage Service (OSS) buckets and their rules are equally distributed between the primary and secondary data centers. Protection groups are created for applications to implement DR.

Geo-disaster recovery supports the following scenarios: cross-cloud DR and cross-region DR. Cross-cloud DR is implemented between two cloud instances. The primary and secondary sites are two independent cloud instances deployed in different locations and use independent account systems. Users must authorize accounts individually to log on to the two cloud instances. Cross-region DR is implemented between two regions. The primary and secondary sites are two regions of a single cloud instance and use the same account system. DR from a central region to a general region and from a general region to a general region are supported in the cross-region DR scenario.

Architecture of cross-cloud DR



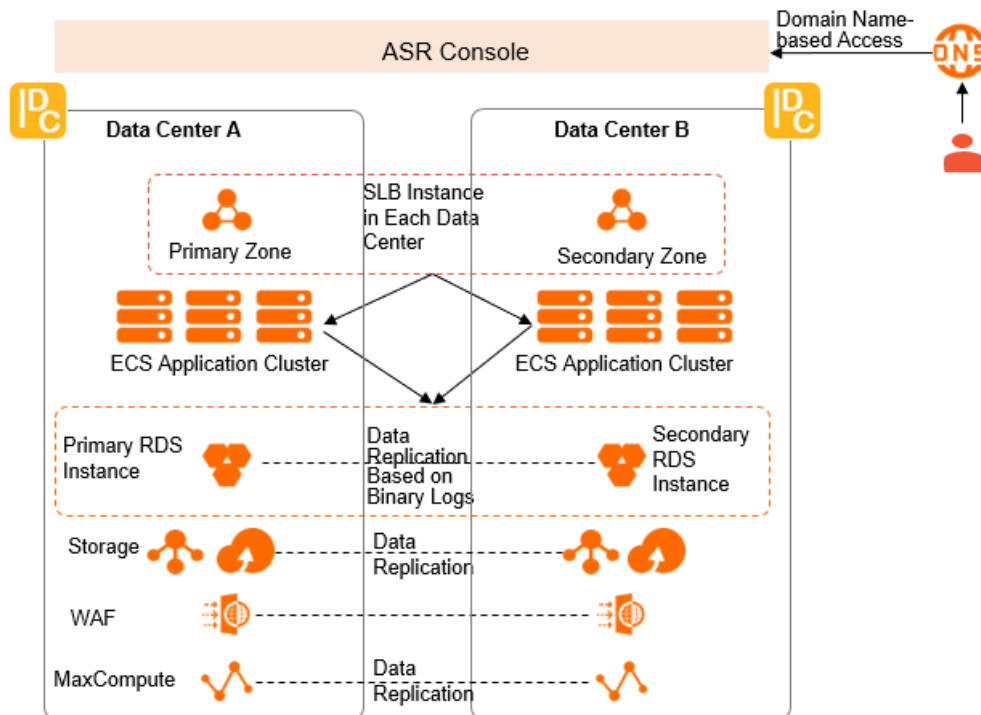
Architecture of cross-region DR



Zone-disaster recovery

Zone-disaster recovery refers to two independent, mutually backed-up data centers within the same zone. When an exception occurs in the primary data center, the secondary data center takes over services by using Apsara Stack Resilience (ASR).

Architecture of two data centers-based zone-disaster recovery

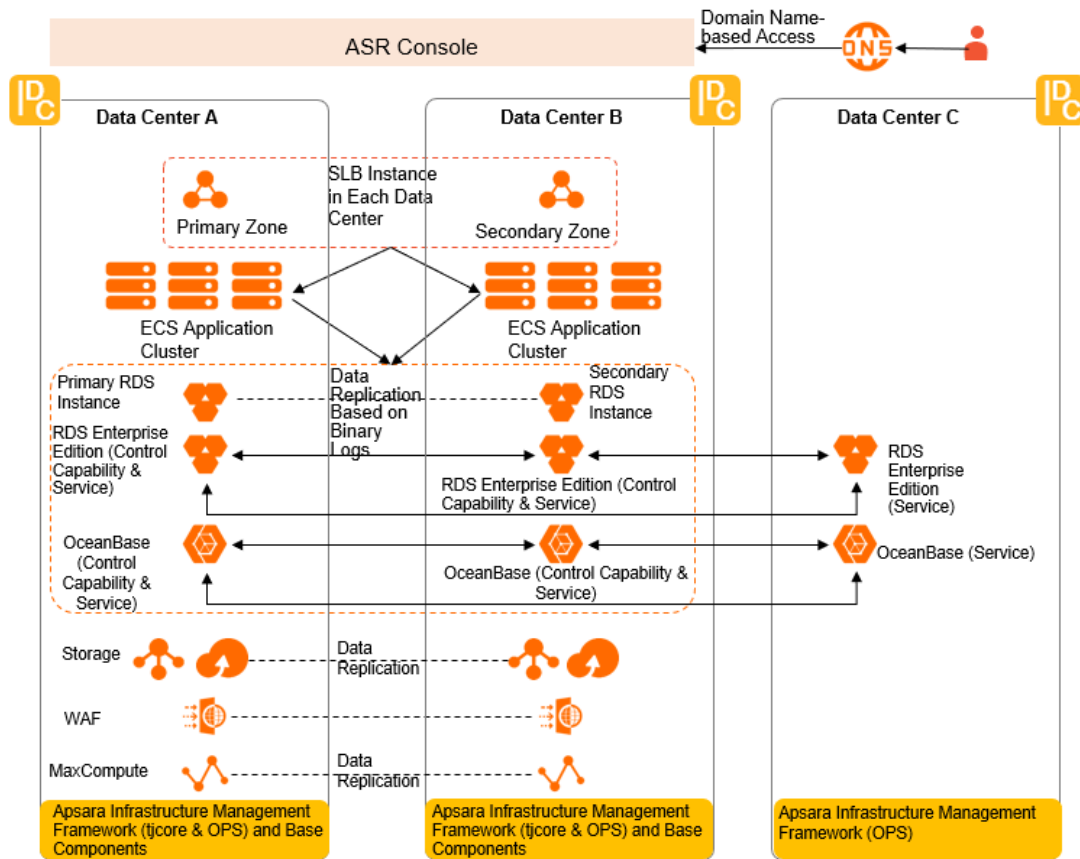


Note Data Center A is the primary data center. Data Center B is the secondary data center.

You can use domain names to access cloud services deployed in the primary and secondary data centers. The domain names of cloud services do not change when services are failed over to the secondary data center. You do not need to remodel your applications. This simplifies application development, makes cloud services easy to use, and allows you to focus on business development.

You can add a third data center based on the architecture of two data centers-based zone-disaster recovery and deploy distributed databases to ensure zero data loss and zero recovery point objective (RPO) in the finance industry. In the architecture of three data centers-based zone-disaster recovery, the DR mechanism for services in active-standby or active-active mode remains unchanged. Failovers are implemented based on the policies used in the architecture of two data centers-based zone-disaster recovery.

Architecture of three data centers-based zone-disaster recovery



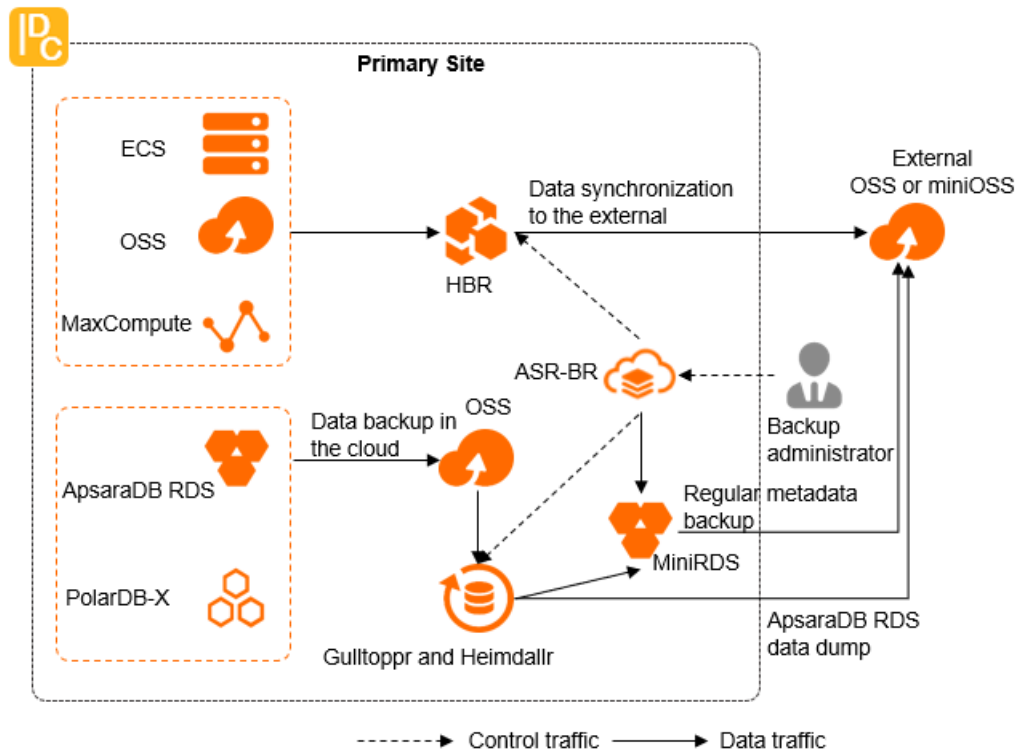
HBR

Apsara Stack HBR support the following two architecture modes: local backup and cross-region backup.

- Local backup

Data is backed up from a primary site to an OSS-compliant storage resource on a regular basis. You must deploy Apsara Stack Resilience for Backup and Recovery (ASR-BR) and related services in the primary site. This architecture reduces deployment costs.

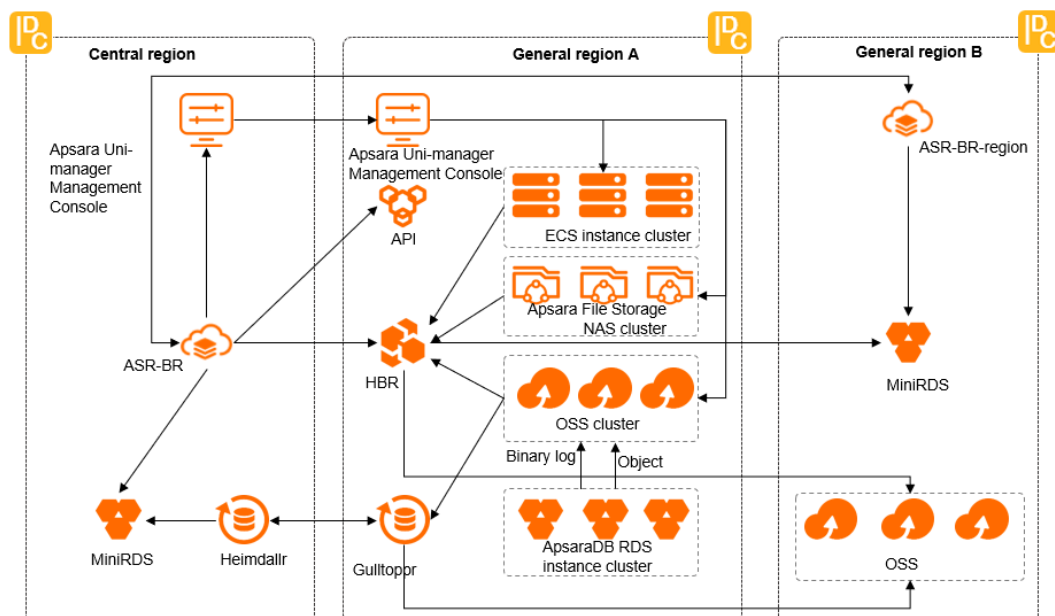
Architecture of local backup



- Cross-region backup

In the multi-region architecture, the data of a region is backed up to another region.

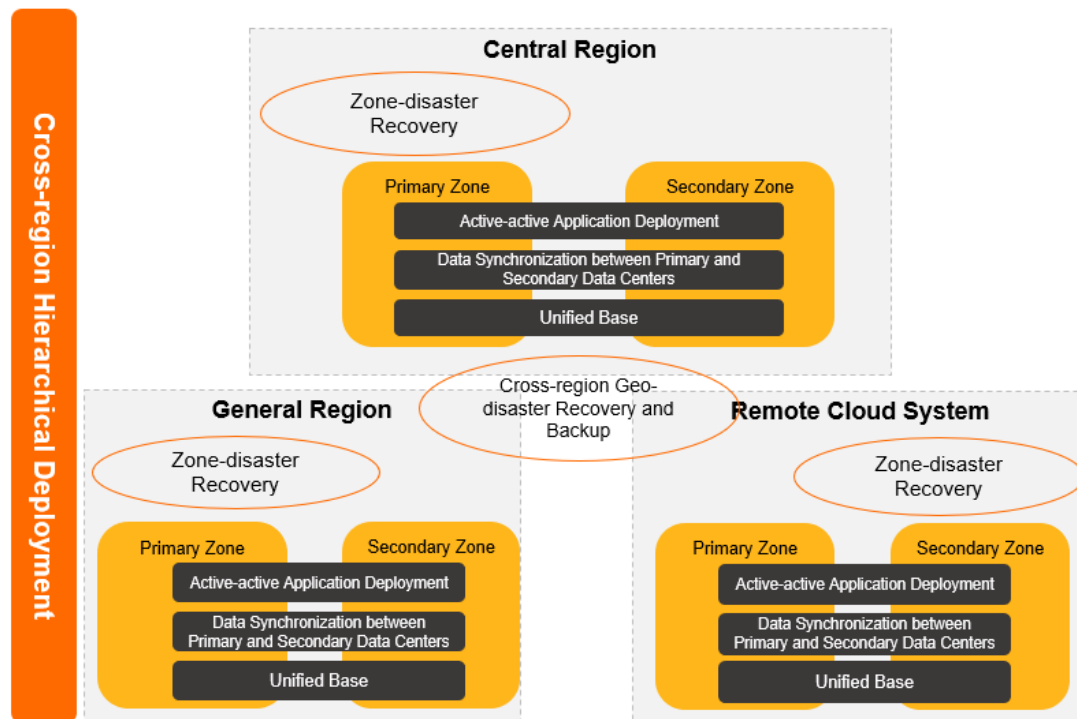
Architecture of cross-region backup



Hybrid networking of multi-region deployment and DR

Apsara Stack Enterprise supports zone-disaster recovery, cross-region DR, and cross-region backup in the multi-region scenario. This can satisfy the DR requirements of a variety of industries.

Hybrid networking architecture of multi-region deployment and DR



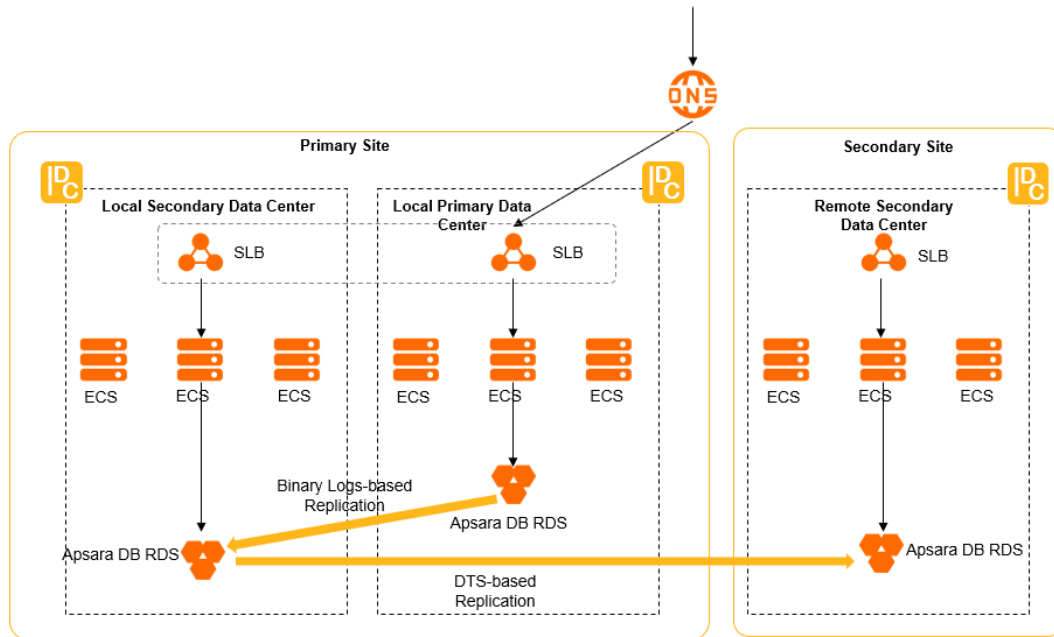
Three data centers across two regions

In the DR solution of three data centers across two regions, two regions include the local and remote regions. Three data centers include the local primary data center, local secondary data center, and the remote secondary data center. This solution can provide high disaster backup capabilities. Cloud services supported by the DR solution include RDS and OSS.

- DR solution for RDS deployed in three data centers across two regions

RDS is independently deployed at each of the primary and secondary sites. Two data centers are deployed at the primary site. RDS is independently deployed in each of the two data centers. The primary and secondary RDS instances deployed at the primary site are used to implement zone-disaster recovery. The primary and secondary sites are used to implement geo-disaster recovery. DR is implemented between two cloud instances by RDS instance.

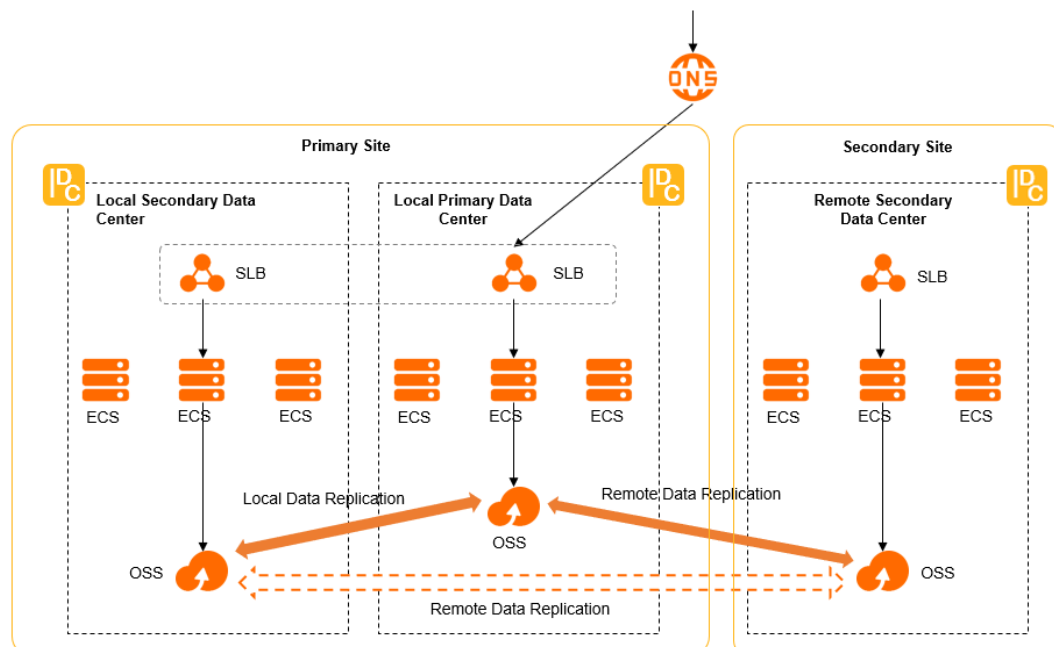
Architecture of RDS deployed in three data centers across two regions



- DR solution for OSS deployed in three data centers across two regions

OSS is independently deployed at each of the primary and secondary sites. Two data centers are deployed at the primary site. OSS is independently deployed in each of the two data centers. The two data centers at the primary site are used to implement zone-disaster recovery. DR is implemented between two data centers by OSS bucket. The primary and secondary sites are used to implement geo-disaster recovery. DR is implemented between two cloud instances by OSS bucket.

Architecture of OSS deployed in three data centers across two regions



The DR solution of three data centers across two regions enhances the business continuity of customer systems. If the primary data center at the primary site fails, the BDR administrator initiates a failover plan in the ASR console. The primary OSS bucket is failed over to the local secondary data center with a single click to ensure business continuity. If both data centers at the primary site fail, the BDR administrator initiates a failover plan in the Apsara Stack Resilience for Disaster Recovery (ASR-DR) console. Then, OSS protection groups are failed over to the secondary site to ensure business continuity. After the primary site is recovered, a reverse data replication tunnel is created between the primary and secondary OSS buckets. This way, incremental data is synchronized from the secondary OSS bucket at the secondary site to the primary bucket at the primary site. After the data is synchronized, start a fallback plan to fail back the traffic to the primary site.

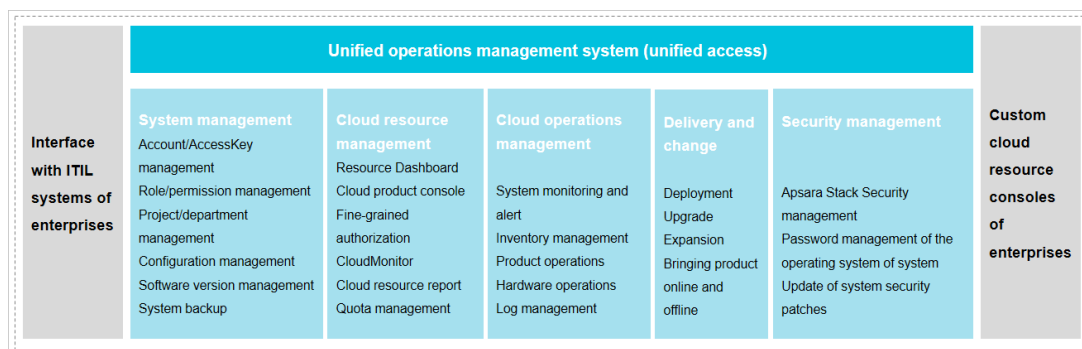
1.2.5. Centralized operations management and automated O&M capability

Apsara Stack provides a centralized management portal. You can configure different management permissions for different roles.

OpenAPI enables you to manage O&M tasks and customize your cloud resource console. Apsara Stack can be synchronized and integrated with the existing Information Technology Infrastructure Library (ITIL) systems of enterprises.

The centralized management system of Apsara Stack provides the same O&M and operations experience as that of Alibaba Cloud.

Centralized O&M management



1.2.6. Rolling upgrade

Apsara Stack supports rolling upgrade and patching. The rolling upgrade process does not interrupt your business. During rolling upgrade, only a few nodes are updated at a time. The rolling upgrade process repeats until all nodes in a cluster are upgraded to the new version.

1.2.7. OpenAPI

Apsara Stack provides a wide range of SDKs and RESTful APIs on the OpenAPI platform.

OpenAPI provides flexible access to a variety of Apsara Stack services. You can also use OpenAPI to obtain the basic control information of Apsara Stack and integrate Apsara Stack with your centralized control system.

1.3. Product architecture

1.3.1. Types of private cloud architectures

This topic describes the following types of private cloud architectures: cloud native architecture and integrated cloud architecture.

- Cloud native architecture

Cloud native architecture was developed from an Internet-based open architecture built on a distributed system framework. Cloud native architecture was originally used for big data and web applications and later used to provide a range of basic services.

- Integrated cloud architecture

Integrated cloud architecture focuses on virtualized computing services. Integrated cloud architecture is a breakthrough from the traditional computing architecture developed by OpenStack and has become the most popular choice for private cloud architecture.

Alibaba Cloud adopts a cloud native architecture and is built on proprietary operating systems, distributed technologies and products of Alibaba Cloud. Apsara Stack uses a single architecture for a variety of deployment environments to support all cloud products. This architecture provides a full set of enterprise-class services, accessibility, disaster recovery and backup capabilities, and self-developed and controllable capabilities.

1.3.2. System architecture

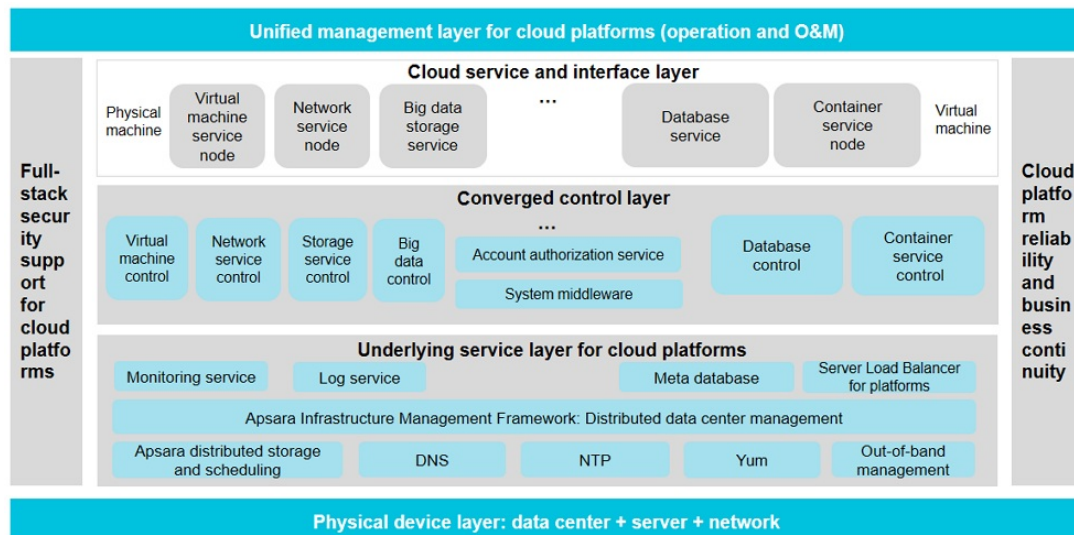
This topic describes the system architecture, logical architecture, and features of each module.

As shown in [System architecture of Apsara Stack](#), the system architecture of Apsara Stack consists of the following layers:

- Physical device layer: includes hardware devices for cloud computing, such as physical data centers, servers, and network.
- Underlying service layer: provides services for upper-layer applications based on the underlying physical environment.
- Converged control layer: provides unified scheduling for upper-layer applications or services by using the converged control architecture.
- Cloud service and interface layer: provides centralized management and O&M for virtual machines and physical machines through converged service node management, and uses the OpenAPI platform to provide centralized API management and support custom development.
- Centralized management layer for cloud platforms: provides centralized O&M management.

Apsara Stack also provides full-stack security support and guarantees the reliability of cloud platforms and business continuity.

System architecture of Apsara Stack

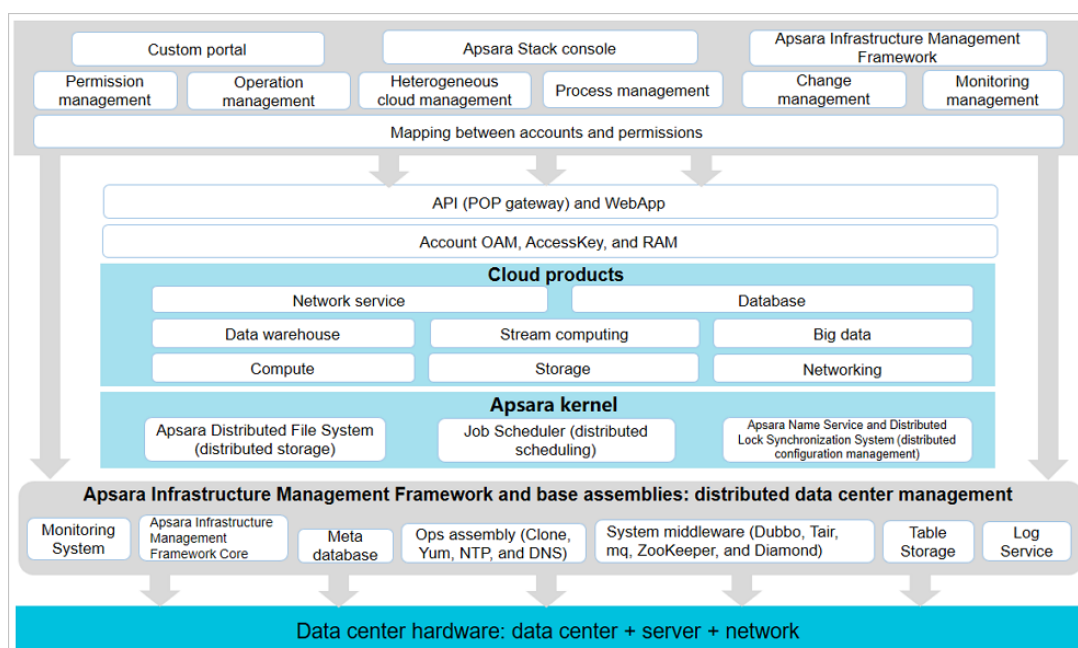


Logical architecture

Apsara Stack virtualizes the computing and storage capabilities of physical servers and network devices to implement virtual computing, distributed storage, and software-defined networks as well as provide ApsaraDB and big data processing. Apsara Stack also provides the supporting capabilities of underlying IT services for your applications, and can be interconnected with your existing account systems and monitoring operations systems. The logical architecture of Apsara Stack has the following characteristics:

- The hardware infrastructure consists of data centers, servers, and network devices. The server supports the x86 processor.
- A variety of cloud services are provided based on the Apsara kernel (distributed engine).
- All cloud services are required to comply with a unified API framework, security system, and O&M and management system (accounts, authorization, monitoring, and logs).
- A consistent user experience is guaranteed across all services.

Logical architecture of Apsara Stack



1.3.3. Network architecture

1.3.3.1. Network architecture overview

Apsara Stack adopts the flat Clos network architecture. The network architecture isolates the service plane from the out-of-band management plane, and supports linear extension and load sharing of switches.

The network architecture contains two logical sections: business service section and integrated access section, as shown in [Logical sections](#).

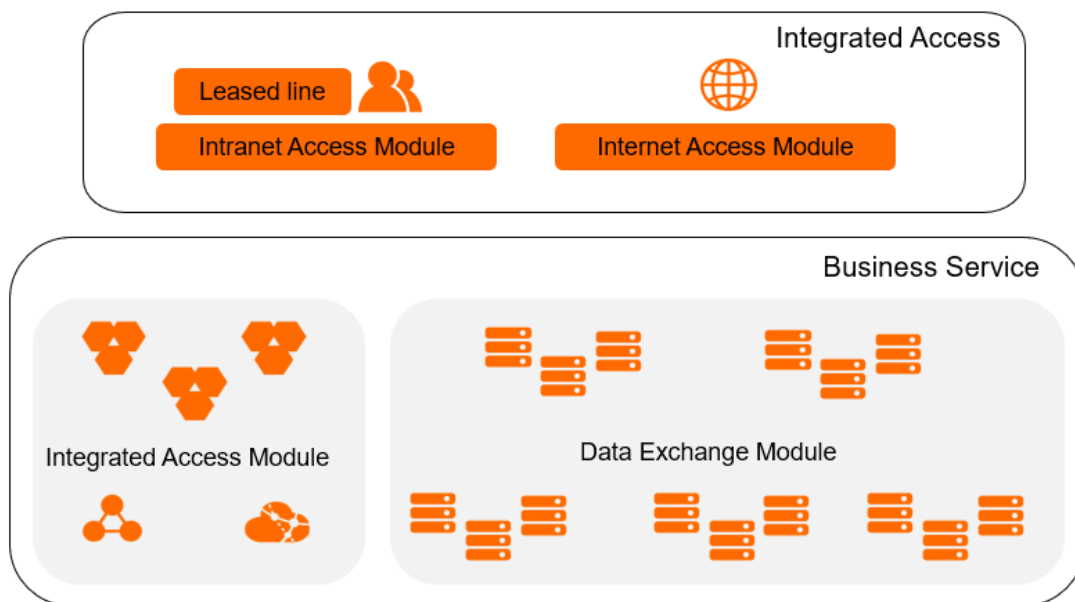
- Business service section

This section provides networks for all cloud services. All cloud service systems exchange internal traffic in this section. This is the core section of Apsara Stack networks.

- Integrated access section

The integrated access section is an extension of the business service section, and provides a channel for user management, and the access to Apsara Stack networks by using Internet and user networks. This section can be tailored based on actual deployment requirements.

Logical sections



The following table describes roles and features of switches in each module.

Role	Module	Description
Inter-connection Switch (ISW)	Internet access module	ISW is an egress switch and provides access to Internet service providers (ISPs) or user backbone networks.

Role	Module	Description
Customer Access Switch (CSW)	Internal network access module	CSW facilitates access to user internal backbone networks, including access to Virtual Private Cloud (VPC) by using leased lines. It performs route distribution and interaction between the internal network and the Internet.
Distribution Switch (DSW)	Data exchange module	DSW is a core switch to connect each Access Switch (ASW).
ASW	Data exchange module	ASW provides access to Elastic Compute Service (ECS) and is uplinked with the core switch DSW.
LVS Switch (LSW)	Integrated access module	LSW provides access to cloud services, such as VPC and Server Load Balancer (SLB).

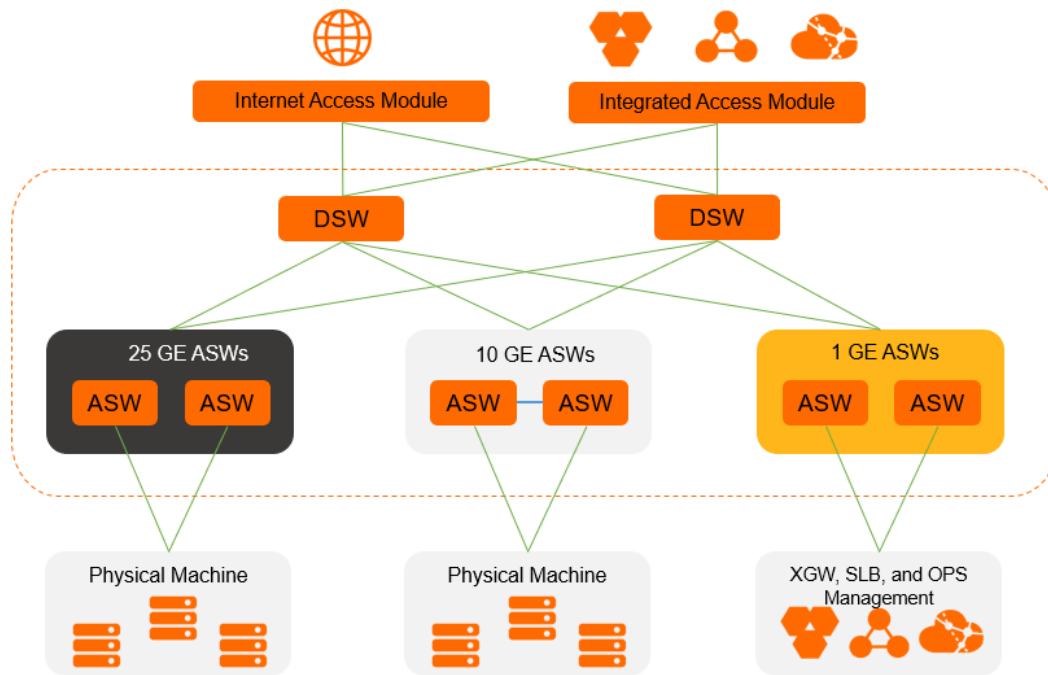
1.3.3.2. Business service section

This topic describes the data exchange and integrated service modules that are contained in the business service section.

- Data exchange module

The data exchange module has a typical layer-2 Clos architecture that consists of DSWs and ASWs. Each ASW pair forms a stack as a leaf node. This node can select data exchange models that have different applicable scopes based on network sizes. All cloud service servers are uplinked with devices on the ASW stacks. ASWs are connected to DSWs by using External Border Gateway Protocol (EBGP). The DSWs are isolated from each other. The data exchange module is connected to other modules by using EBGP. This module receives the Internet routes from ISWs and releases the CIDR blocks of cloud services to the ISWs.

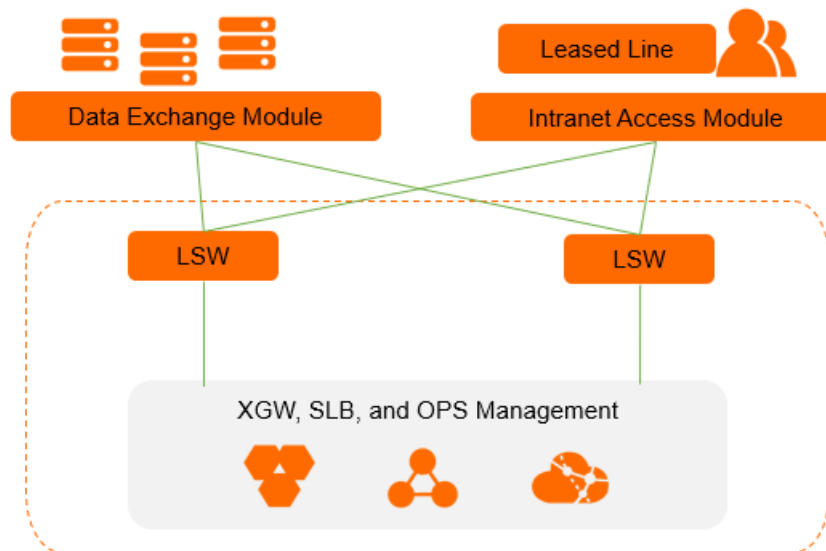
Data exchange module



- Integrated service module

Each cloud service server (XGW, SLB, or OPS) is connected to two LSWs. These servers exchange routing information by using Open Shortest Path First (OSPF). Two LSWs exchange routing information between each other by using Internal Border Gateway Protocol (IBGP). LSWs exchange routing information with DSWs and CSWs by using EBGp.

Integrated service module



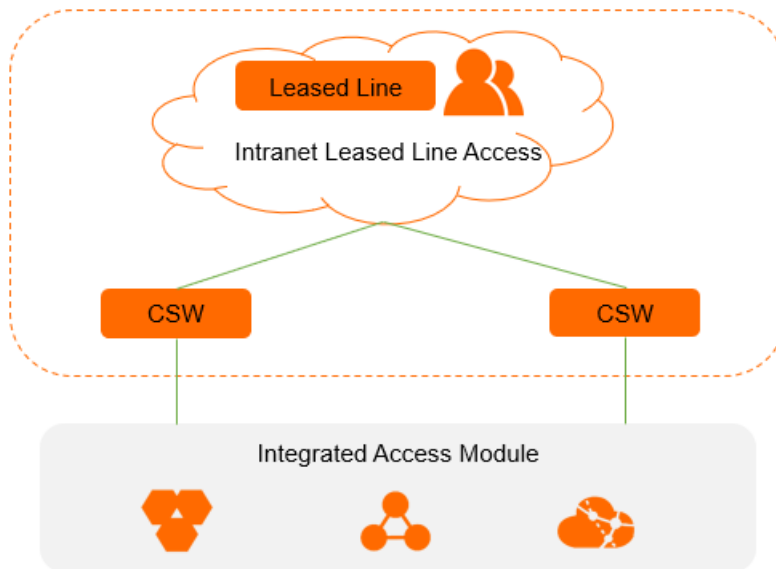
1.3.3.3. Integrated access section

This topic describes the internal network access module and Internet access module that are contained in the integrated access section.

- Internal network access module

Two CSWs provide internal users with access to VPCs and general cloud services. CSWs establish mappings between internal users and VPCs to route these users to different VPCs. Different groups are isolated from each other on CSWs. For access to general cloud services, CSWs are connected to the integrated service module by using EBGP and allow direct access to all resources in the business service section.

Internal network access module

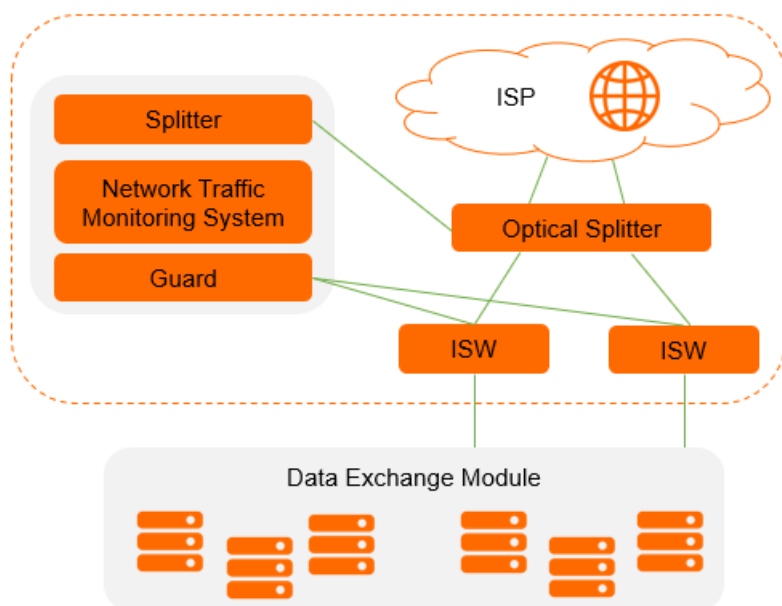


- Internet access module

The Internet access module consists of two ISWs. This module facilitates the access to Internet service providers (ISPs) or user public backbone networks and performs route distribution and interaction between the internal network and Internet. The two ISWs use IBGP to back up routes between each other. ISWs can use static routing or EBGP to uplink with ISPs or user public backbone networks based on actual conditions. The link bandwidth is determined by the Alibaba Cloud network size of users and the bandwidth of their public backbone networks. We recommend that ISWs use BGP to connect with multiple ISPs to improve reliability. Each ISP has two 10 GE lines. The Internet access module uses EBGP to exchange routes with the data exchange module. The Internet access module releases relevant Internet routes to the data exchange module and receives internal cloud service routes that are sent by the data exchange module to implement the interaction between the internal network and Internet.

The Internet access module is connected to the Alibaba Cloud security protection system in one-arm mode. Traffic that is transmitted from the Internet to cloud networks is diverted by an optical splitter to Network Traffic Monitoring System. If Network Traffic Monitoring System detects malicious traffic, it releases the corresponding routes by using Apsara Stack Security to divert the malicious traffic to Apsara Stack Security for scrubbing. The scrubbed traffic is then injected back into the Internet access module.

Internet access module

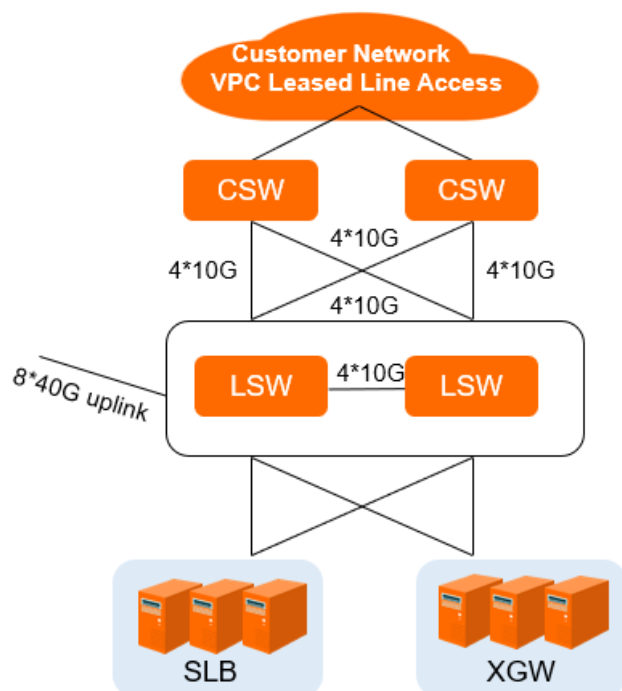


1.3.3.4. VPC leased line access

The leased line access solution of VPC allows users to control their own virtual networks, such as selecting their own CIDR blocks and configuring route tables and gateways. Users can also connect their VPC to a traditional data center by using leased lines or VPN connections to create a custom network environment. This enables smooth migration of applications to the cloud.

Each cloud service server (XGW and SLB) is connected to two LSWs. These servers exchange routing information by using OSPF. Two LSWs use IBGP to exchange routing information to each other. LSWs exchange routing information with CSWs by using EBGP.

VPC leased line access



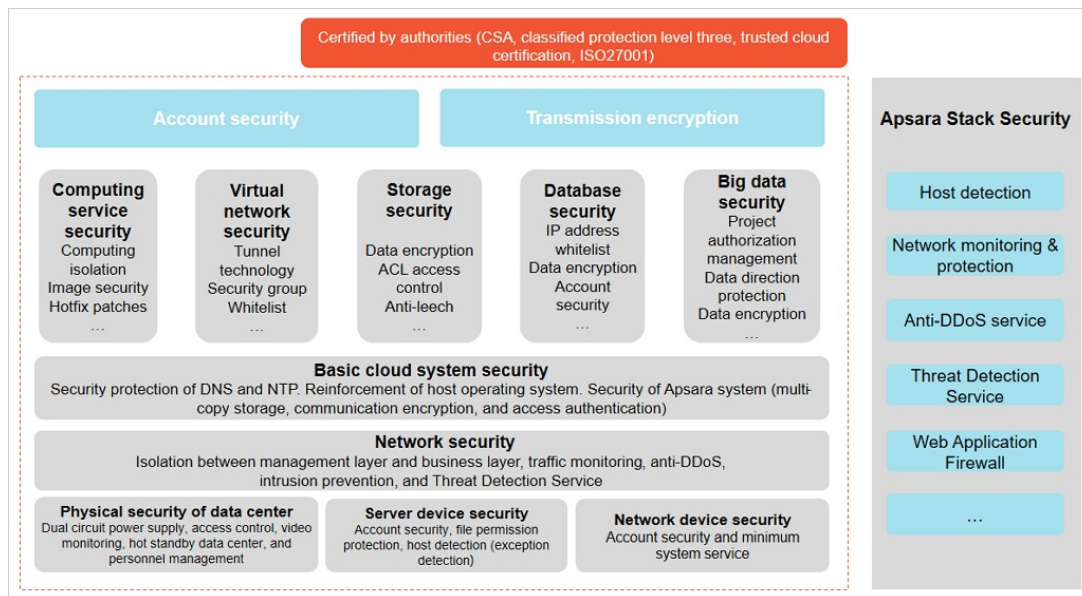
1.3.4. Security architecture

Apsara Stack provides comprehensive security capabilities from underlying communication protocols to upper-layer applications to guarantee the security of your access and data.

Access to all consoles in Apsara Stack is allowed only with HTTPS certificates. Apsara Stack provides a comprehensive role authorization mechanism to guarantee a secure and controllable access to resources in multi-tenant mode. It supports a variety of security roles, such as a security administrator, system administrator, and security auditor.

In addition, Apsara Stack V3.0 and later introduce Apsara Stack Security services to provide you with a multi-level and integrated cloud security solution.

Hierarchical security architecture of Apsara Stack



1.3.5. Base modules

The Apsara Stack base consists of three module types, all of which provide support for the deployment and O&M of the cloud platform.

Base modules

Module		Feature description
	YUM	The installation package. Software repositories are deployed in the initial installation stage to install the operating system and deploy application packages such as the Apsara system and Elastic Compute Service (ECS), and dependent modules of Apsara Stack on hosts.
	Clone	The machine cloning service.

Module		Feature description
OPS modules	NTP	The clock source service. NTP is deployed on hosts of Apsara Stack to synchronize time from the standard NTP clock source to other hosts.
	DNS	The domain name resolution service. DNS provides forward and reverse resolution of domain names for the internal Apsara Stack environment. It runs a bind instance on each of the two OPS servers and uses Keepalived to provide high availability services. When one server fails, the other server automatically takes over its work.
Base middleware	Dubbo	The distributed remote procedure call (RPC) service.
	Tair	The cache service.
	Message Queue (MQ)	The message queue service.
	ZooKeeper	The distributed collaboration service.
	Diamond	The configuration management service.
	SchedulerX	The scheduled task service.
Basic modules of the base	Apsara Infrastructure Management Framework	The data center management system.
	Monitoring System	The data center monitoring system.
	OTS-inner	The table storage service.
	SLS-inner	The log service of the cloud platform.
	Metadatabase	The metadatabases.
	POP	The Apsara Stack OpenAPI platform.

Module		Feature description
	OAM	The account system.
	RAM	The authentication and authorization system.
	WebApps	The service that provides support for the Apsara Uni-manager Operations Console.

1.4. Product panorama

Apsara Stack offers a wide range of services to meet the diverse requirements of different users.

Architecture

Apsara Stack provides a wide variety of basic virtual resources, such as virtual computing, virtual networking, and virtual scheduling. The main products include Elastic Compute Service (ECS), Virtual Private Cloud (VPC), Server Load Balancer (SLB), Container Service, Auto Scaling, Resource Orchestration Service (ROS), NAT Gateway, VPN Gateway, Elastic IP Address (EIP), and Apsara Stack DNS.

Storage services

Apsara Stack provides various storage services for different storage objects. The main services include Object Storage Service (OSS).

Database services

Apsara Stack provides a variety of database engines that can communicate with each other. The main services include ApsaraDB RDS, KVStore for Redis, ApsaraDB for MongoDB, Data Management (DMS), Tablestore, AnalyticDB for PostgreSQL, and PolarDB-X.

Big data services

Apsara Stack provides a variety of features such as big data analysis, application, and visualization to ensure data value is used to its fullest potential. The main services include MaxCompute, DataWorks, DataHub, Realtime Compute, and Apsara Big Data Manager (ABM).

Artificial intelligence services

Apsara Stack provides a machine learning algorithm platform based on the distributed computing engine developed by Alibaba Cloud. The main services include Machine Learning Platform for AI (PAI).

Security services

Apsara Stack provides comprehensive protection from underlying communication protocols all the way up to upper-layer applications to ensure access and data security. The supported service is Apsara Stack Security. Apsara Stack also provides services such as secure hosting of keys and cryptographic operations. The supported service is Key Management Service (KMS).

1.5. Scenarios

Apsara Stack provides flexible and scalable industrial solutions for customers of different scales and sectors.

Apsara Stack can create customized solutions based on the unique business traits of different sectors such as industry, agriculture, transportation, government, finance, and education to provide users with end-to-end products and services. This topic describes the following scenarios.

City Brain

Urban management is a field that involves one of the largest volumes of data in China. This marks the transition of governmental information from a closed-flow model to an open-flow online model. Urban data has a greater value as it has more time and larger space to flow. Cloud computing becomes urban infrastructure. Data becomes a new means of production and strategic resources. AI technology becomes the nerve center of a smart city. All of these together form the City Data Brain.

City Brain has the following values and features:

- A breakthrough of urban governance mode. City Brain uses urban data as resources to improve government management capabilities, resolve prominent issues of urban governance, and implement an intelligent, intensive, and humane form of governance.
- A breakthrough of urban service mode. City Brain provides more accurate and convenient services for enterprises and individuals, makes urban public services more efficient, and saves more public resources.
- A breakthrough of urban industrial development. City Brain lays down an industrial AI layout, takes open urban data as an important fundamental resource, drives the development of industries, and promotes the transformation and upgrade of traditional industries.

Alibaba Finance Cloud

Alibaba Finance Cloud is an industrial cloud that serves financial organizations, such as banks, security agencies, insurance companies, and finance. It relies on a cluster of independent data centers to provide cloud products that meet the regulatory requirements of the People's Bank of China, China Banking Regulatory Commission (CBRC), China Securities Regulatory Commission (CSRC), and China Insurance Regulatory Commission (CIRC). It also provides more professional and comprehensive services for financial customers. Enterprises can build Alibaba Finance Cloud independently or with Alibaba Cloud. Alibaba Finance Cloud meets the requirements of large and medium-sized financial organizations for independent cloud data centers that are completely physically isolated. It can also provide cloud computing and big data platforms for data centers of customers.

Alibaba Finance Cloud has the following values and features:

- Independent resource clusters
- Stricter data center management
- Better disaster recovery capability
- Stricter requirements for network security isolation
- Stricter access control
- Compliance with the security supervision requirements and compliance requirements of banks
- Dedicated security operation, compliance, and solution teams of the Alibaba Finance Cloud sector
- Dedicated account managers and cloud architects of Alibaba Finance Cloud
- Stricter user access mechanism

1.6. Compliance security solution

1.6.1. Overview

This topic describes the laws and regulations related to network security and Apsara Stack security compliance solutions.

On June 1, 2017, the Cybersecurity Law of the People's Republic of China was officially implemented and provided clear guidelines for classified protection compliance. To help you align with the provisions for classified protection compliance, Alibaba Cloud uses its technical expertise on Apsara Stack Security products to build a classified protection compliance ecosystem. Alibaba Cloud works with its cooperative assessment agencies and security consulting providers based worldwide to offer one-stop classified protection assessment services. Apsara Stack offers complete attack protection, data auditing, encryption, and security management to simplify classified protection compliance assessment and ensure you pass quickly.

1.6.2. Interpretation of key points

Network and communication security

Clause interpretation

- Divide the network into different security zones based on server roles and importance.
- Set access control policies at the security zone boundary between the internal network and the Internet, which must be configured on specific ports.
- Deploy intrusion prevention measures at the network boundary to prevent against and record intrusion behaviors.
- Record and audit user behavior logs and security events in the network.

Coping strategies

- We recommend that you use Apsara Stack Virtual Private Cloud (VPC) and security groups to divide networks into different security zones and perform appropriate access control.
- You can use Web Application Firewall (WAF) to prevent network intrusion.
- You can use the log feature to record, analyze, and audit user behavior logs and security events.
- If the system is frequently threatened by DDoS attacks, you can use Anti-DDoS Pro to filter and scrub abnormal traffic.

Device and computing security

Clause interpretation

- Adhere to basic security operations such as performing record and audit operations and discouraging account sharing.
- Ensure the security of the system layer by using necessary security measures and prevent servers from intrusions.

Coping strategies

- You can audit the operations on servers and data, and create an independent account for each of O&M personnel to avoid account sharing.
- You can use Server Guard to implement complete vulnerability management, baseline check, and intrusion prevention on servers.

Application and data security

Clause interpretation

- An application directly implements specific business and does not have the standard characteristics of networks and systems. The features of most applications, such as identity authentication, access control, and operation audit, are difficult to replace with third-party products.
- Encryption is the most effective method to keep data integrated and confidential except security prevention methods at other levels.
- Remote data backup is one of the most important requirements to distinguish the Multi-level Protection Scheme (MLPS) Level III from Level II. It is the most fundamental technical safeguard for business continuity.

Coping strategies

- At the beginning of the application development, application features such as identity authentication, access control, and security audit must be taken into consideration.
- For online systems, you can add features such as account authentication, user permission identification, and log audit to satisfy classified protection requirements.
- For data security, HTTPS can be used to ensure that data remains encrypted during transmission.
- For data backup, we recommend that you use ApsaraDB RDS geo-disaster recovery instances to automatically back up data, and manually synchronize database backup files to Apsara Stack servers that are deployed in other regions.

Security management

Clause interpretation

- Security policies, regulations, and management personnel are important foundations to ensure sustainable security. Policies guide the security direction. Regulations specify the security process. Management personnel shoulder the security responsibilities.
- Classified protection requirements provide a methodology and best practices. You can continuously construct and manage security based on the classified protection methodology.

Coping strategies

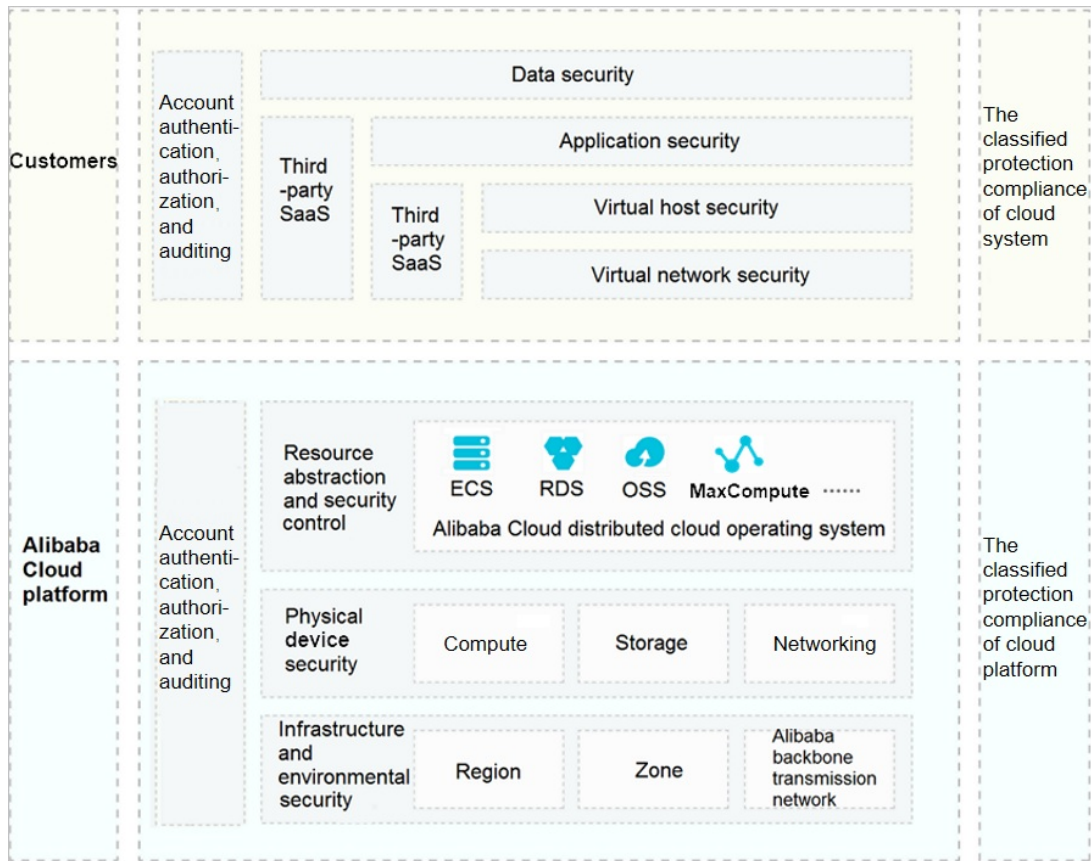
- The customer management staff can arrange, prepare, and implement the security policies and regulations as well as allocate management personnel based on the actual conditions of enterprises, and then formulate specialized documents.
- For the technical methods required in vulnerability management, we recommended that you use Apsara Stack Server Guard to detect and handle the vulnerabilities of cloud systems in a timely manner.

1.6.3. Cloud-based classified protection compliance

Shared compliance responsibilities

The Alibaba Cloud platform and the cloud tenant systems are classified and assessed respectively. You can use the assessment conclusions of the Alibaba Cloud platform when assessing the tenant systems.

Shared compliance responsibilities



Alibaba Cloud provides the following contents:

- Classified protection filing certification of the Alibaba Cloud platform
- Key pages of the Alibaba Cloud assessment report
- Sales license of Apsara Stack Security
- Description of partial assessment items of Alibaba Cloud

More details about shared responsibilities are as follows:

- Alibaba Cloud is the unique cloud service provider in China that participates in and passes the pilot demonstration of cloud computing classified protection standards. Public Cloud and E-Government Cloud pass the filing and assessment of the third level of classified protection. Finance Cloud passes the filing and assessment of the fourth level of classified protection.
- According to the regulatory authority, you can use the assessment conclusions of physical security, partial network security, and security management for the classified protection assessment of the tenant systems on Alibaba Cloud, and Alibaba Cloud can provide supporting details.
- With the complete security technology, management architecture, and protection system of Apsara Stack Security, Alibaba Cloud platform makes it easy for tenants to pass the classified protection assessment.

Classified protection compliance ecology

Current conditions of cloud-based classified protection are as follows:

- Most tenants do not know classified protection.
- Most tenants do not know how to start with classified protection.
- Most tenants are not good at communicating with supervision authorities.

- Security systems lag behind business development.

Alibaba Cloud establishes Classified Protection Compliance Ecology to provide one-stop classified protection compliance solutions for cloud-based systems to quickly pass classified protection assessment.

Classified protection compliance ecology



Work division of classified protection:

- Alibaba Cloud: integrates capabilities of service agencies and provides security products
- Consulting firm: provides technical support and consulting services in the whole process
- Assessment agency: provides assessment services
- Public security organ: reviews filing and supervises services

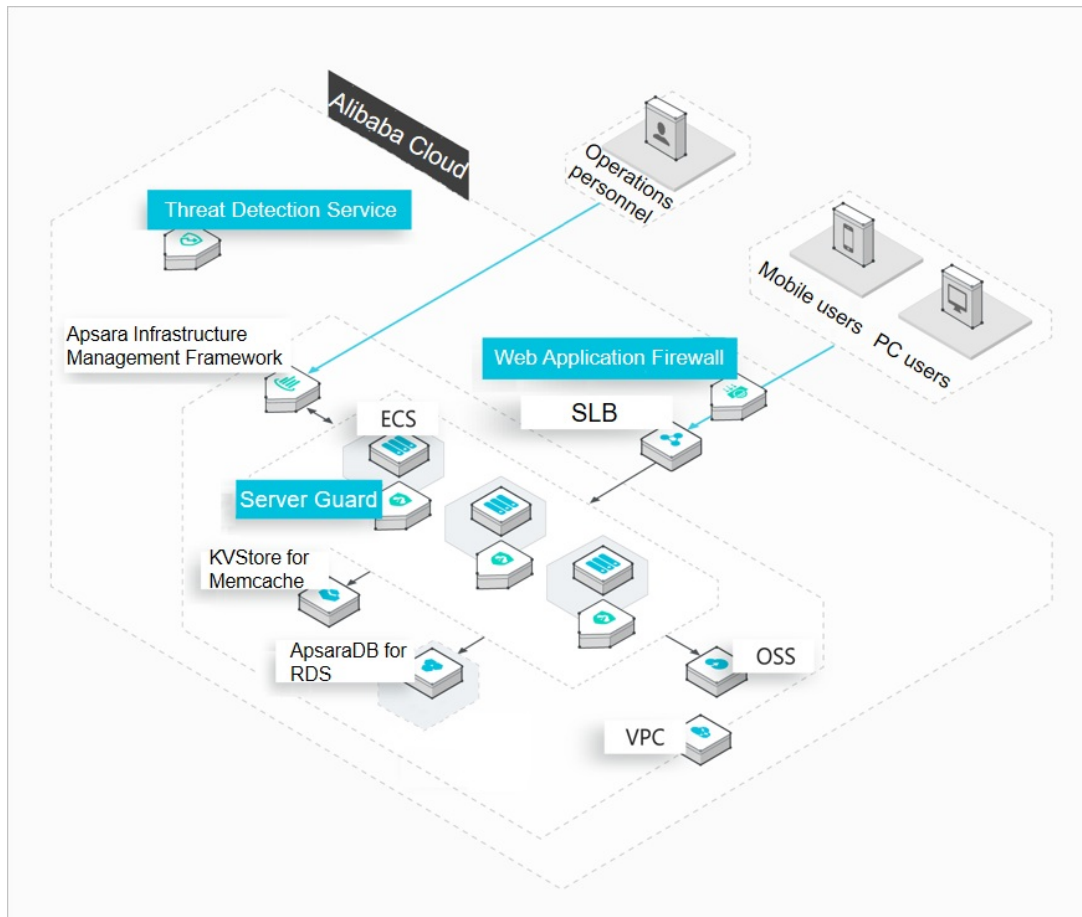
1.6.4. Classified protection implementation process

	Operating unit	Alibaba Cloud	Consulting or assessment agency	Public security organ
System rating	Determine the class of security protection and write rating report	Coordinate the third party agency to provide counseling services for operating unit	Counseling the operating unit to prepare the rating materials and organize expert review (level three)	None
System filing	Prepare and present the filing materials to the local public security organ	Coordinate the third party agency to provide counseling services for operating unit	Counseling the operating unit to prepare the filing materials and to issue filing	None
Construction rectification	Construct the security technology and management system in line with class requirements	Provide the obligatory security products and services that meet the class requirements	Counseling the operating unit to carry out system security reinforcement and develop security management regulation	The local public security organ reviews and accepts the filing materials
Rating assessment	Prepare for and accept the assessment from the assessment agency	Provide the cloud service provider's security qualification and the proof that the cloud platform has passed the classified protection	The assessment agency assesses the system class conformity	None
Supervision & inspection	Accept the regular inspection of public security organ	None	None	Supervise and inspect the operating unit to carry out the class protection work

1.6.5. Security compliance architecture

With the security compliance architecture, Alibaba Cloud can fast connect to Apsara Stack Security, quickly improve the security, and comply with basic technical requirements for classified protection at minimal security costs.

Security compliance architecture



Basic requirements of classified protection are as follows:

- Physical and environmental security: includes data center power supply, temperature and humidity control, and prevention of wind, rain, and lightning. You can use the assessment conclusions of Alibaba Cloud.
- Network and communication security: includes network architecture, boundary protection, access control, intrusion prevention, and communication encryption.
- Device and computing security: includes intrusion prevention, malicious code prevention, identity authentication, access control, centralized control, and security auditing.
- Application and data security: includes security auditing, data integrity, and data confidentiality.

1.6.6. Solution benefits

One-stop assessment service of classified protection

Select high-performance consulting and assessment partners to provide one-stop compliance support throughout, allowing the operators to achieve significant cost savings.

- Eliminates multi-level communication and work redundancy to help the operators reduce investment.
- Improves efficiency by shortening the assessment cycle to as short as two weeks.
- Alibaba Cloud provides best practices of security and compliance on the cloud.

A complete security protection system

With a complete Apsara Stack Security architecture, operators can locate corresponding products on Alibaba Cloud, rectify non-conformances, and meet all requirements of classified protection.

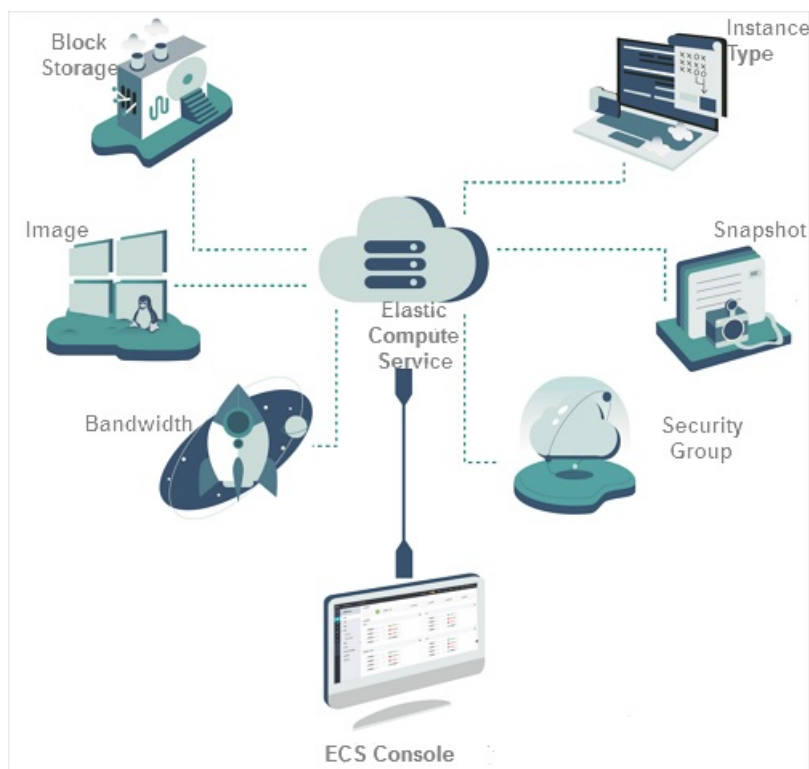
2. Elastic Compute Service (ECS)

2.1. What is ECS?

Elastic Compute Service (ECS) is a computing service that features elastic processing capabilities. Compared with physical servers, ECS instances are more user-friendly and can be managed more efficiently. You can create instances, resize disks, and add or release any number of ECS instances at any time based on your business needs.

An ECS instance is a virtual computing environment that contains the most basic components of computers such as the CPU, memory, and storage. Users perform operations on ECS instances. Instances are core components of ECS, and operations can be performed on instances through the ECS console. Other resources, such as block storage, images, and snapshots, can only be used after they are integrated with ECS instances. For more information, see [ECS components](#).

ECS components



2.2. Benefits

ECS provides higher availability, security, and elasticity than services provided by other server vendors and traditional Internet data centers (IDCs).

High availability

ECS employs stringent IDC standards, server access standards, and O&M standards to ensure data reliability and high availability at the infrastructure and instance levels.

Apsara Stack provides the following support services:

- Industry and ecosystem partners to help you build a more advanced and stable architecture and

ensure service continuity.

- Diverse training services to help you implement high availability from the business end to the underlying basic service end.

Security

Security and stability are two of the primary concerns for any cloud service users. Alibaba Cloud has received a host of international information security certifications that demand strict confidentiality of user data and user privacy protection, including ISO 27001 and Multi-Tier Cloud Security (MTCS).

- **Apsara Stack Virtual Private Cloud (VPC) uses simple configurations** to increase the flexibility, scalability, and stability of your business.
- **You can connect your own IDC to Apsara Stack VPC** by using leased lines to build a hybrid cloud. You can use a variety of hybrid cloud architectures to provide network services and robust networking.
- **VPCs are more stable and secure.**
 - VPCs allow you to divide, configure, and manage your network.
 - VPCs provide traffic isolation and attack isolation to protect your services against cyber attacks. You can establish a first line of defense against malicious attacks and traffic by building your business in a VPC.

VPCs provide a stable, secure, controllable, and fast-deliverable network environment. The capability and architecture of the VPC hybrid cloud bring the technical advantages of cloud computing to enterprises in traditional industries not engaged in cloud computing.

Elasticity

Elasticity is a key benefit of cloud computing.

- **Elastic computing**
 - **Vertical scaling**

Vertical scaling is the process of changing the configurations of ECS instances. In a traditional IDC, it can be difficult to change the configurations of individual servers. However, in Apsara Stack, you can change the configurations of your ECS instances based on the volume of your business.
 - **Horizontal scaling**

Horizontal scaling allows you to scale the quantity of resources for applications. A traditional IDC may not be able to immediately provide sufficient resources for online gaming or live video streaming applications during peak hours. The elasticity of cloud computing makes it possible to provide as many resources as required during peak hours. When the load returns to normal levels, you can release redundant resources to reduce operation costs.

The combination of ECS vertical and horizontal elasticity and Auto Scaling enables you to scale resources up and down by specified quantities as scheduled or against business loads.
- **Elastic storage**

Apsara Stack provides elastic storage. In a traditional IDC, you must add servers to increase the storage space. However, the number of servers that you can add is limited. In Apsara Stack, you can purchase storage products such as cloud disks at any time.
- **Elastic network**

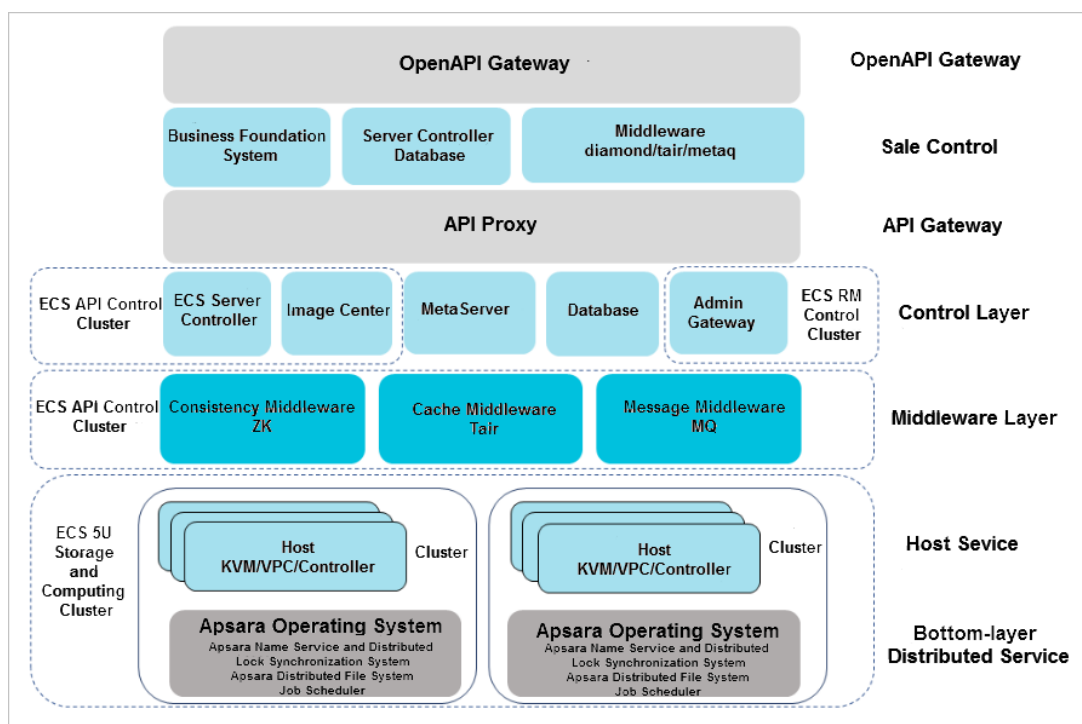
Apsara Stack provides network elasticity. When you purchase Apsara Stack VPCs, you can configure the VPCs in the same way as you configure your IDCs. In addition, VPCs have the following benefits: interconnection between data centers, separate secure domains in data centers, and flexible network configurations and planning within a VPC.

In conclusion, Apsara Stack provides elastic computing, storage, networking as well as business architecture planning. You can use Apsara Stack to build your business portfolio in any way.

2.3. Architecture

ECS is built on the distributed Apsara system developed by Alibaba Cloud. Individual ECS instances are virtualized by using KVM while storage is implemented on Apsara Distributed File System.

ECS architecture



Architecture description

Component	Description
Apsara Name Service and Distributed Lock Synchronization System	A basic module that provides services related to distributed consensus in Apsara Stack. As a key distributed coordination system of Apsara Stack, this module provides three types of basic services: distributed lock services, subscription and notification services, and lightweight metadata storage services.
Apsara Distributed File System	A distributed storage system developed by Alibaba Cloud. Since 2017, hundreds of clusters and hundreds of thousands of storage nodes that use Apsara Distributed File System have been deployed in production environments. Apsara Distributed File System manages several exabytes of disk space.

Component	Description
Job Scheduler	A distributed resource scheduler that manages and allocates resources in the distributed systems.
Scheduling Process	API > Business layer > Control system > Host service.
OpenAPI Gateway	Provides services such as authentication and request forwarding.
Business Foundation System	Creates and releases instances and snapshot policies, processes sales requests, and provides APIs to users.
API Proxy	Forwards requests to services that are deployed in the region specified by region_id.
Server Controller database	Stores control data and status data.
Server Controller	Schedules storage, network, and computing resources.
Tair	Provides cache services for Server Controller.
Zookeeper	Provides the distributed lock service for Server Controller.
MQ	Provides message queues for the statuses of virtual machines.
Image Center	Provides image management services such as import and copy.
MetaSever	Provides metadata management services for ECS instances.
Host service	Provides services such as KVM for computing virtualization, VPC for network virtualization, and control through interaction with Libvirt.
AG(Admin Gateway)	Functions as the bastion host used to log on to an NC during O&M management.
ECS Decider	Determines the NC on which to deploy ECS instances.

2.4. Features

2.4.1. Instances

2.4.1.1. Overview

An ECS instance is a virtual machine that contains basic computing components such as CPU, memory, operating system, and network. All configurations of an ECS instance can be modified and customized to suit your needs. You can log on to the Apsara Uni-manager Management Console to manage resources and configure the environments of your ECS instances.

2.4.1.2. Instance families

An Elastic Compute Service (ECS) instance is the smallest unit that can provide compute capabilities and services for your business. Compute capabilities vary based on instance types. ECS instances are categorized into different instance families based on the business scenarios for which they are suited.

 **Note** The instance families and instance types described in this topic are for reference only. The physical server where an instance is hosted determines the instance type.

Instance families based on Intel x86 servers

Instance family	Feature	Scenario
s6, shared standard instance family	- Is an instance family in which all instances are I/O optimized by default. - Offers multiple CPU-to-memory ratios. - Uses the latest DDR4 memory. - Supports standard SSDs and ultra disks.	- Web applications and small and medium-sized websites - Development environments, servers, code repositories, microservices, and testing and staging environments - Lightweight databases and caches - Lightweight enterprise applications and integrated application services
n4, shared general purpose instance family	- Offers a CPU-to-memory ratio of 1:2. - Uses 2.5 GHz Intel® Xeon® E5-2680 v3 (Broadwell) processors. - Uses the latest DDR4 memory. - Is an instance family in which all instances are I/O optimized by default.	- Small and medium-sized web servers - Batch processing - Advertising services - Distributed analysis
mn4, shared balanced instance family	- Offers a CPU-to-memory ratio of 1:4. - Uses 2.5 GHz Intel® Xeon® E5-2680 v3 (Broadwell), Intel® Xeon® E5-2680 v4 (Haswell), Intel® Xeon® E5-2682 v4 (Broadwell), or Intel® Xeon® E5-2650 v2 (Haswell) processors. - Uses the latest DDR4 memory. - Is an instance family in which all instances are I/O optimized by default.	- Small and medium-sized web servers - Batch processing - Advertising services - Distributed analysis - Hadoop clusters

Instance family	Feature	Scenario
xn4, shared compact instance family	- Offers a CPU-to-memory ratio of 1:1. - Uses 2.5 GHz Intel® Xeon® E5-2680 v4 (Haswell) or Intel® Xeon® E5-2682 v4 (Broadwell) processors. - Uses the latest DDR4 memory. - Is an instance family in which all instances are I/O optimized by default.	- Minisite web applications - Small databases - Development and testing environments - Code storage servers
e4, shared memory optimized instance family	- Offers a CPU-to-memory ratio of 1:8. - Uses 2.5 GHz Intel® Xeon® E5-2680 v4 (Broadwell), Intel® Xeon® E5-2680 v3 (Broadwell), Intel® Xeon® E5-2650 v2 (Haswell), or Intel® Xeon® E5-2682 v4 (Broadwell) processors. - Is an instance family in which all instances are I/O optimized by default.	Applications that involve large numbers of memory operations, queries, and computations, such as Cache, Redis, search applications, and in-memory databases
n4v2, shared general purpose instance family	- Supports IPv6. - Offers a CPU-to-memory ratio of 1:2. - Uses 2.5 GHz Intel® Xeon® E5-2680 v3 (Broadwell) processors. - Uses the latest DDR4 memory. - Is an instance family in which all instances are I/O optimized by default.	- Small and medium-sized web servers - Batch processing - Advertising services - Distributed analysis

Instance family	Feature	Scenario
mn4v2, shared balanced instance family	• Supports IPv6. • Offers a CPU-to-memory ratio of 1:4. • Uses 2.5 GHz Intel® Xeon® E5-2680 v3 (Broadwell), Intel® Xeon® E5-2680 v4 (Haswell), Intel® Xeon® E5-2682 v4 (Broadwell), or Intel® Xeon® E5-2650 v2 (Haswell) processors. • Uses the latest DDR4 memory. • Is an instance family in which all instances are I/O optimized by default.	• Small and medium-sized web servers • Batch processing • Advertising services • Distributed analysis • Hadoop clusters
xn4v2, shared compact instance family	• Supports IPv6. • Offers a CPU-to-memory ratio of 1:1. • Uses 2.5 GHz Intel® Xeon® E5-2680 v4 (Haswell) or Intel® Xeon® E5-2682 v4 (Broadwell) processors. • Uses the latest DDR4 memory. • Is an instance family in which all instances are I/O optimized by default.	• Minisite web applications • Small databases • Development and testing environments • Code storage servers
e4v2, shared memory optimized instance family	• Supports IPv6. • Offers a CPU-to-memory ratio of 1:8. • Uses 2.5 GHz Intel® Xeon® E5-2680 v4 (Broadwell), Intel® Xeon® E5-2680 v3 (Broadwell), Intel® Xeon® E5-2650 v2 (Haswell), or Intel® Xeon® E5-2682 v4 (Broadwell) processors. • Is an instance family in which all instances are I/O optimized by default.	Applications that involve large numbers of memory operations, queries, and computations, such as Cache, Redis, search applications, and in-memory databases

Instance family	Feature	Scenario
t5, burstable instance family	• Uses 2.5 GHz Intel® Xeon® processors. • Uses the DDR4 memory. • Offers multiple CPU-to-memory ratios. • Provides baseline CPU performance and is burstable, but limited by accumulated CPU credits. • Offers a balance of computing, memory, and network resources. • Supports only virtual private clouds (VPCs).	• Web application servers • Lightweight applications and microservices • Applications for development and testing services
sn1ne, compute optimized instance family with enhanced network performance	• Offers a CPU-to-memory ratio of 1:2. • Provides ultra-high packet forwarding rates. • Uses 2.5 GHz Intel® Xeon® E5-2682 v4 (Broadwell) or E5-2680 v4 (Haswell) processors for consistent computing performance. • Provides high network performance based on large computing capacity.	• Scenarios where large volumes of packets are received and transmitted, such as on-screen video comments and telecom data forwarding • Web frontend servers • Frontend servers of massively multiplayer online (MMO) games • Data analysis, batch processing, and video encoding • High-performance scientific and engineering applications
g7, general purpose instance family	• Is an instance family in which all instances are I/O optimized. • Supports standard SSDs and ultra disks. • Offers a CPU-to-memory ratio of 1:4. • Provides ultra-high packet forwarding rates. • Provides high storage I/O performance based on large computing capacity. • Provides high network performance based on large computing capacity.	• Scenarios where large volumes of packets are received and transmitted, such as on-screen video comments and telecom data forwarding • Enterprise-level applications of various types and sizes • Websites and application servers • Game servers • Small and medium-sized database systems, caches, and search clusters • Data analysis and computing • Compute clusters and memory-intensive data processing

Instance family	Feature	Scenario
g6, general purpose instance family	• Offloads a large number of virtualization features to dedicated hardware with the use of the SHENLONG architecture to provide predictable and consistent ultra-high performance and reduce virtualization overheads. • Is an instance family in which all instances are I/O optimized. • Supports standard SSDs and ultra disks. • Provides high storage I/O performance based on large computing capacity. • Offers a CPU-to-memory ratio of 1:4. • Allows you to enable or disable Hyper-Threading. • Provides ultra-high packet forwarding rates. • Uses 2.5 GHz Intel® Xeon® Platinum 8269CY (Cascade Lake) processors that deliver a maximum turbo frequency of 3.2 GHz for consistent computing performance. • Provides high network performance based on large computing capacity.	• Scenarios where large volumes of packets are received and transmitted, such as on-screen video comments and telecom data forwarding • Enterprise-level applications of various types and sizes • Websites and application servers • Game servers • Small and medium-sized database systems, caches, and search clusters • Data analysis and computing • Compute clusters and memory-intensive data processing
g5, general purpose instance family	• Is an instance family in which all instances are I/O optimized. • Supports standard SSDs and ultra disks. • Offers a CPU-to-memory ratio of 1:4 • Provides ultra-high packet forwarding rates. • Uses 2.5 GHz Intel® Xeon® Platinum 8163 (Skylake) or 8269CY (Cascade Lake) processors for consistent computing performance. • Provides high network performance based on large computing capacity.	• Scenarios where large volumes of packets are received and transmitted, such as on-screen video comments and telecom data forwarding • Enterprise-level applications of various types and sizes • Small and medium-sized database systems, caches, and search clusters • Data analysis and computing • Compute clusters and memory-intensive data processing

Instance family	Feature	Scenario
sn2ne, general purpose instance family with enhanced network performance	- Offers a CPU-to-memory ratio of 1:4 - Provides ultra-high packet forwarding rates. - Uses 2.5 GHz Intel® Xeon® E5-2682 v4 (Broadwell) or E5-2680 v4 (Haswell) processors for consistent computing performance. - Provides high network performance based on large computing capacity.	- Scenarios where large volumes of packets are received and transmitted, such as on-screen video comments and telecom data forwarding - Enterprise-level applications of various types and sizes - Small and medium-sized database systems, caches, and search clusters - Data analysis and computing - Compute clusters and memory-intensive data processing
r6, memory optimized instance family	- Offloads a large number of virtualization features to dedicated hardware with the use of the SHENLONG architecture to provide predictable and consistent ultra-high performance and reduce virtualization overheads. - Is an instance family in which all instances are I/O optimized. - Supports standard SSDs and ultra disks. - Provides high storage I/O performance based on large computing capacity. - Provides ultra-high packet forwarding rates. - Uses 2.5 GHz Intel® Xeon® Platinum 8269CY (Cascade Lake) processors that deliver a maximum turbo frequency of 3.2 GHz for consistent computing performance. - Offers a CPU-to-memory ratio of 1:8. - Allows you to enable or disable Hyper-Threading. - Provides high network performance based on large computing capacity.	- Scenarios where large volumes of packets are received and transmitted, such as on-screen video comments and telecom data forwarding - High-performance and in-memory databases - Data analysis, data mining, and distributed memory caching - Hadoop clusters, Spark clusters, and other memory-intensive enterprise applications

Instance family	Feature	Scenario
r5, memory optimized instance family	- Is an instance family in which all instances are I/O optimized. - Supports standard SSDs and ultra disks. - Offers a CPU-to-memory ratio of 1:8. - Provides ultra-high packet forwarding rates. - Uses 2.5 GHz Intel® Xeon® Platinum 8163 (Skylake) or Intel® Xeon® Platinum 8269CY (Cascade Lake) processors for consistent computing performance. - Provides high network performance based on large computing capacity.	- Scenarios where large volumes of packets are received and transmitted, such as on-screen video comments and telecom data forwarding - High-performance and in-memory databases - Data analysis, data mining, and distributed memory caching - Hadoop clusters, Spark clusters, and other memory-intensive enterprise applications
se1ne, memory optimized instance family with enhanced network performance	- Offers a CPU-to-memory ratio of 1:8. - Provides ultra-high packet forwarding rates. - Uses 2.5 GHz Intel® Xeon® E5-2682 v4 (Broadwell) or Platinum 8163 (Skylake) processors for consistent computing performance. - Provides high network performance based on large computing capacity.	- Scenarios where large volumes of packets are received and transmitted, such as on-screen video comments and telecom data forwarding - High-performance and in-memory databases - Data analysis, data mining, and distributed memory caching - Hadoop clusters, Spark clusters, and other memory-intensive enterprise applications
se1, memory optimized instance family	- Provides consistent computing performance. - Offers a CPU-to-memory ratio of 1:8. - Uses 2.5 GHz Intel® Xeon® E5-2682 v4 (Broadwell) or E5-2680 v4 (Haswell) processors. - Uses the latest DDR4 memory. - Provides high network performance based on large computing capacity. - Is an instance family in which all instances are I/O optimized by default.	- Cache and Redis - Search applications - In-memory databases - Databases with high I/O requirements, such as Oracle and MongoDB - Hadoop clusters - Computing scenarios where consistent performance is required, such as large-volume data processing

Instance family	Feature	Scenario
ebmg5s, general purpose ECS Bare Metal Instance family with enhanced network performance	• Supports standard SSDs and ultra disks. • Offers a CPU-to-memory ratio of 1:4. • Uses 2.5 GHz Intel® Xeon® Platinum 8163 (Skylake) processors that have 96 vCPUs and a maximum turbo frequency of 2.7 GHz. • Provides high network performance with a packet forwarding rate of 4,500 Kpps. • Supports only VPCs. • Provides dedicated hardware resources and physical isolation.	• Workloads that require direct access to physical resources or scenarios that require a license to be bound to the hardware • Third-party virtualization (including Xen and KVM) and AnyStack (including OpenStack and ZStack) • Containers (including Docker, Clear Containers, and Pouch) • Enterprise-level applications such as large and medium-sized databases • Video encoding
ebmg5, general purpose ECS Bare Metal Instance family	• Offers a CPU-to-memory ratio of 1:4. • Uses 2.5 GHz Intel® Xeon® Platinum 8163 (Skylake) processors that have 96 vCPUs and a maximum turbo frequency of 2.9 GHz. • Provides high network performance with a packet forwarding rate of 4,500 Kpps. • Supports standard SSDs and ultra disks.	• Deployment of Apsara Stack services such as OpenStack and ZStack • Deployment of services such as Docker containers • Scenarios where large volumes of packets are received and transmitted, such as on-screen video comments and telecom data forwarding • Enterprise-level applications of various types and sizes • Large and medium-sized databases, caches, and search clusters • Data analysis and computing • Compute clusters and memory-intensive data processing
i2, instance family with local SSDs	• Uses high-performance local NVMe SSDs that deliver high IOPS, high I/O throughput, and low latency. • Offers a CPU-to-memory ratio of 1:8, which is designed for high-performance databases. • Uses 2.5 GHz Intel® Xeon® Platinum 8163 (Skylake) processors. • Provides high network performance based on large computing capacity.	• OLTP and high-performance relational databases • NoSQL databases such as Cassandra and MongoDB • Search scenarios that use solutions such as Elasticsearch

Instance family	Feature	Scenario
d1, big data instance family	• Uses high-capacity local SATA HDDs that deliver high throughput and a maximum of 17 Gbit/s bandwidth among instances. • Offers a CPU-to-memory ratio of 1:4, which is designed for big data scenarios. • Uses 2.5 GHz Intel® Xeon® E5-2682 v4 (Broadwell) processors. • Provides high network performance based on large computing capacity.	• Hadoop MapReduce, HDFS, Hive, and HBase • Spark in-memory computing and MLlib • Industries such as Internet and finance that need to compute, store, and analyze big data • Elasticsearch and logging
d2, big data instance family	• Uses high-capacity local SATA HDDs that deliver high throughput and a maximum of 10 Gbit/s bandwidth among instances. • Uses 2.5 GHz Intel® Xeon® Platinum 8163 (Skylake) processors. • Provides high network performance based on large computing capacity.	• Hadoop MapReduce, HDFS, Hive, and HBase • Spark in-memory computing and MLlib • Industries such as Internet and finance that need to compute, store, and analyze big data • Elasticsearch and logging
sccg5ib, general purpose SCC instance family	• Offers a CPU-to-memory ratio of 1:8. • Provides ultra-high packet forwarding rates. • Uses 2.5 GHz Intel® Xeon® Platinum 8163 (Skylake) processors for consistent computing performance. • Supports 100 Gbit/s InfiniBand networks that deliver ultra-high bandwidth and ultra-low latency.	• Data analysis and computing • Artificial intelligence computing • Manufacturing simulation • High-performance compute clusters • Genetic analysis • Pharmaceutical analysis
scch5ib, SCC instance family with high clock speed	• Offers a CPU-to-memory ratio of 1:6. • Provides ultra-high packet forwarding rates. • Uses 3.1 GHz Intel® Xeon® Gold 6149 (Skylake) processors for consistent computing performance. • Supports 100 Gbit/s InfiniBand networks that deliver ultra-high bandwidth and ultra-low latency.	• Data analysis and computing • Artificial intelligence computing • Manufacturing simulation • High-performance compute clusters • Genetic analysis • Pharmaceutical analysis

Instance family	Feature	Scenario
c6, compute optimized instance family	• Offloads a large number of virtualization features to dedicated hardware with the use of the SHENLONG architecture to provide predictable and consistent ultra-high performance and reduce virtualization overheads. • Is an instance family in which all instances are I/O optimized. • Supports standard SSDs and ultra disks. • Provides high storage I/O performance based on large compute capacity. • Allows you to enable or disable Hyper-Threading. • Offers a CPU-to-memory ratio of 1:2. • Provides ultra-high packet forwarding rates. • Uses 2.5 GHz Intel® Xeon® Platinum 8269CY (Cascade Lake) processors that deliver a maximum turbo frequency of 3.2 GHz for consistent computing performance. • Provides high network performance based on large computing capacity.	• Scenarios where large volumes of packets are received and transmitted, such as on-screen video comments and telecom data forwarding • Web frontend servers • Frontend servers of MMO games • Data analysis, batch processing, and video encoding • High-performance scientific and engineering applications
sn1, compute optimized instance family	• Offers a CPU-to-memory ratio of 1:2. • Uses 2.5 GHz Intel® Xeon® E5-2682 v4 (Broadwell) or E5-2680 v3 (Haswell) processors for consistent computing performance. • Provides high network performance based on large computing capacity.	• Web frontend servers • Frontend servers of MMO games • Data analysis, batch processing, and video encoding • High-performance scientific and engineering applications

Instance family	Feature	Scenario
sn2, general purpose instance family	- Offers a CPU-to-memory ratio of 1:4. - Uses 2.5 GHz Intel® Xeon® E5-2682 v4 (Broadwell) or E5-2680 v3 (Haswell) processors for consistent computing performance. - Provides high network performance based on large computing capacity.	- Enterprise-level applications of various types and sizes - Small and medium-sized database systems, caches, and search clusters - Data analysis and computing - Compute clusters and memory-intensive data processing

Instance family	Feature	Scenario
f1, FPGA-accelerated compute optimized instance family	- Uses Intel® ARRIA® 10 GX 1150 FPGAs. - Offers a CPU-to-memory ratio of 1:7.5. - Uses 2.5 GHz Intel® Xeon® E5-2682 v4 (Broadwell) processors. - Provides high network performance based on large computing capacity.	- Deep learning and inference - Genomics research - Financial analysis - Image transcoding - Computational workloads such as real-time video processing and security management
f3, FPGA-accelerated compute optimized instance family	- Uses Xilinx Virtex UltraScale+ VU9P. - Offers a CPU-to-memory ratio of 1:4. - Uses 2.5 GHz Intel® Xeon® Platinum 8163 (Skylake) processors. - Provides high network performance based on large computing capacity.	- Deep learning and inference - Genetic computation - Video encoding and decoding. - Chip prototype verification - Database acceleration

Instance family	Feature	Scenario
gn5, GPU-accelerated compute optimized instance family	- Uses NVIDIA P100 GPUs. - Offers multiple CPU-to-memory ratios. - Uses high-performance NVMe SSDs. - Uses 2.5 GHz Intel® Xeon® E5-2682 v4 (Broadwell) processors. - Provides high network performance based on large computing capacity.	- Deep learning - Scientific computing applications, such as fluid dynamics, finance, genomics, and environmental analysis - Server-side GPU compute workloads such as high-performance computing, rendering, and multi-media encoding and decoding

Instance family	Feature	Scenario
gn4, GPU-accelerated compute optimized instance family	• Uses NVIDIA M40 GPUs. • Offers multiple CPU-to-memory ratios. • Uses 2.5 GHz Intel® Xeon® E5-2680 v4 (Haswell) processors. • Provides high network performance based on large computing capacity.	• Deep learning • Scientific computing applications, such as fluid dynamics, finance, genomics, and environmental analysis • Server-side GPU compute workloads such as high-performance computing, rendering, and multi-media encoding and decoding
ga1, GPU-accelerated visualization and compute optimized instance family	• Uses AMD S7150 GPUs. • Offers a CPU-to-memory ratio of 1:2.5. • Uses 2.5 GHz Intel® Xeon® E5-2682 v4 (Broadwell) processors. • Uses high-performance NVMe SSDs. • Provides high network performance based on large computing capacity.	• Rendering and multi-media encoding and decoding • Machine learning, high-performance computing, and high-performance databases • Server-side workloads that require powerful concurrent floating-point computing capacity
gn5i, GPU-accelerated compute optimized instance family	• Uses NVIDIA P4 GPUs. • Offers a CPU-to-memory ratio of 1:4 • Uses 2.5 GHz Intel® Xeon® E5-2682 v4 (Broadwell) or E5-2680 v4 (Haswell) processors. • Provides high network performance based on large computing capacity.	• Deep learning and inference • Server-side GPU computational workloads such as multimedia encoding and decoding
gn5e, GPU-accelerated compute optimized instance family	• Is an instance family in which all instances are I/O optimized. • Uses 2.5 GHz Intel® Xeon® Platinum 8163 (Skylake) processors for consistent computing performance. • Uses NVIDIA P4 GPUs. • Provides high network performance based on large computing capacity.	• Deep learning and inference • Video and image processing, such as noise reduction, encoding, and decoding

Instance family	Feature	Scenario
gn6i, GPU-accelerated compute optimized instance family	• Is an instance family in which all instances are I/O optimized. • Offers a CPU-to-memory ratio of 1:4. • Uses 2.5 GHz Intel® Xeon® Platinum 8163 (Skylake) processors. • Supports standard SSDs and ultra disks. • Uses the new-generation SHENLONG compute architecture for higher performance. • Uses NVIDIA T4 GPUs that feature: ◦ Innovative NVIDIA Turing architecture ◦ Up to 320 Turing Tensor cores per GPU ◦ Up to 2,560 CUDA cores per GPU ◦ Mixed-precision Tensor cores that support 65 FP16 TFLOPS, 130 INT8 TOPS, and 260 INT4 TOPS ◦ 16 GB memory per GPU (320 GB/s bandwidth) ◦ Provides high network performance based on large computing capacity.	• AI (deep learning and machine learning) inference for computer vision, voice recognition, speech synthesis, natural language processing (NLP), machine translation, and reference systems • Real-time rendering for cloud gaming • Real-time rendering for AR and VR applications • Graphics workstations or overloaded graphics computing • GPU-accelerated databases • High-performance computing

Instance family	Feature	Scenario
gn6v, GPU-accelerated compute optimized instance family	- Is an instance family in which all instances are I/O optimized. - Supports standard SSDs and ultra disks. - Uses NVIDIA T4 GPUs. - Offers a CPU-to-memory ratio of 1:4. - Uses 2.5 GHz Intel® Xeon® Platinum 8163 (Skylake) processors. - Uses NVIDIA V100 GPUs (SXM2-based) that feature: - Innovative NVIDIA Volta architecture 16 GB HBM2 memory per GPU (900 GB/s bandwidth) - Up to 5,120 CUDA cores per GPU - Up to 640 Tensor cores per GPU - Support for up to six NVLink connections and total bandwidth of 300 GB/s (25 GB/s per connection) - Provides high network performance based on large computing capacity.	- Deep learning applications, such as training and inference applications of AI algorithms used in image classification, autonomous vehicles, and speech recognition - Scientific computing applications, such as fluid dynamics, finance, molecular dynamics, and environmental analysis
vgn6i, lightweight GPU-accelerated compute optimized instance family	- Is an instance family in which all instances are I/O optimized. - Supports standard SSDs and ultra disks. - Uses NVIDIA T4 GPUs. - Is equipped with vGPUs that are generated from GPU virtualization with mediated pass-through. - Offers a CPU-to-memory ratio of 1:5. - Uses a variety of processors such as Intel® Xeon® Platinum 8163 (Skylake).	Cloud desktops

Instance family	Feature	Scenario
sccgn6p, GPU-accelerated compute optimized SCC instance family	- Is an instance family in which all instances are I/O optimized. - Offers a CPU-to-memory ratio of 1:8. - Uses 2.5 GHz Intel® Xeon® Platinum 8163 (Skylake) processors for consistent computing performance. - Provides all the features of ECS Bare Metal Instance. - Storage: - Supports standard SSDs and ultra disks. - Supports high-performance Cloud Parallel File System (CPFS). - Supports Apsara File Storage NAS. - Networking: - Supports VPCs equipped with two 25 Gbit/s ports. - Supports 100 Gbit/s InfiniBand networks for low-latency RDMA communication. - Uses NVIDIA V100 GPUs (SXM2-based) that feature: - Innovative NVIDIA Volta architecture 32 GB HBM2 GPU memory - CUDA Cores 5120 - Tensor Cores 640 - GPU memory bandwidth of up to 900 GB/s - Support for up to six NVLink connections and total bandwidth of 300 GB/s (25 GB/s per connection)	- Ultra-large-scale training for machine learning on a distributed GPU cluster - Large-scale high-performance scientific computing and simulations - Large-scale data analysis, batch processing, and video encoding

Instance family	Feature	Scenario

Instance family	Feature	Scenario
Custom instance type ecs.anyshare	- Allows you to create, modify, and delete custom instance types in the Apsara Uni-manager Operations Console. - Allows you to create instances of custom instance types. - Custom instance types are shared instance types.	Scenarios where existing instance types cannot meet the requirements and instance types need to be customized

The following instance families are applicable only in environments that are upgraded from Apsara Stack V2 to V3.

Instance family	Feature	Scenario
n1, shared compute optimized instance family	- Offers a CPU-to-memory ratio of 1:2. - Uses 2.5 GHz Intel® Xeon® E5-2680 v3 (Haswell) processors. - Provides high network performance based on large computing capacity. - Is an instance family in which all instances are I/O optimized. - Supports standard SSDs and ultra disks.	- Small and medium-sized web servers - Batch processing - Distributed analysis - Advertising services
n2, shared general purpose instance family	- Offers a CPU-to-memory ratio of 1:4. - Uses 2.5 GHz Intel® Xeon® E5-2680 v3 (Haswell) processors. - Provides high network performance based on large computing capacity. - Is an instance family in which all instances are I/O optimized. - Supports standard SSDs and ultra disks.	- Medium-sized web servers - Batch processing - Distributed analysis - Advertising services - Hadoop clusters

Instance family	Feature	Scenario
e3, shared memory optimized instance family	• Offers a CPU-to-memory ratio of 1:8. • Uses 2.5 GHz Intel® Xeon® E5-2680 v3 (Haswell) processors. • Provides high network performance based on large computing capacity. • Is an instance family in which all instances are I/O optimized. • Supports standard SSDs and ultra disks.	• Cache and Redis • Search applications • In-memory databases • Databases with high I/O requirements, such as Oracle and MongoDB • Hadoop clusters • Large-volume data processing
c1, generation I instance family	• Uses Intel® Xeon® E5-2420 processors that deliver a minimum clock speed of 1.9 GHz. • Uses the latest DDR3 memory. • Is an instance family in which instances can be I/O optimized or non-I/O optimized. • I/O optimized instances support both standard SSDs and ultra disks. • Non-I/O optimized instances support only basic disks.	Generation I instance types are legacy shared instance types. They are still categorized based on the number of cores (1, 2, 4, 8, and 16 cores) and are not sensitive to instance families.
c2, generation I instance family	• Uses Intel® Xeon® E5-2420 processors that deliver a minimum clock speed of 1.9 GHz. • Uses the latest DDR3 memory. • Is an instance family in which instances can be I/O optimized or non-I/O optimized. • I/O optimized instances support both standard SSDs and ultra disks. • Non-I/O optimized instances support only basic disks.	Generation I instance types are legacy shared instance types. They are still categorized based on the number of cores (1, 2, 4, 8, and 16 cores) and are not sensitive to instance families.

Instance family	Feature	Scenario
m1, generation I instance family	• Uses Intel® Xeon® E5-2420 processors that deliver a minimum clock speed of 1.9 GHz. • Uses the latest DDR3 memory. • Is an instance family in which instances can be I/O optimized or non-I/O optimized. • I/O optimized instances support both standard SSDs and ultra disks. • Non-I/O optimized instances support only basic disks.	Generation I instance types are legacy shared instance types. They are still categorized based on the number of cores (1, 2, 4, 8, and 16 cores) and are not sensitive to instance families.
m2, generation I instance family	• Uses Intel® Xeon® E5-2420 processors that deliver a minimum clock speed of 1.9 GHz. • Uses the latest DDR3 memory. • Is an instance family in which instances can be I/O optimized or non-I/O optimized. • I/O optimized instances support both standard SSDs and ultra disks. • Non-I/O optimized instances support only basic disks.	Generation I instance types are legacy shared instance types. They are still categorized based on the number of cores (1, 2, 4, 8, and 16 cores) and are not sensitive to instance families.
s1, generation I instance family	• Uses Intel® Xeon® E5-2420 processors that deliver a minimum clock speed of 1.9 GHz. • Uses the latest DDR3 memory. • Is an instance family in which all instances are non-I/O optimized. • Supports only basic disks.	Generation I instance types are legacy shared instance types. They are still categorized based on the number of cores (1, 2, 4, 8, and 16 cores) and are not sensitive to instance families.
s2, generation I instance family	• Uses Intel® Xeon® E5-2420 processors that deliver a minimum clock speed of 1.9 GHz. • Uses the latest DDR3 memory. • Is an instance family in which instances can be I/O optimized or non-I/O optimized. • I/O optimized instances support both standard SSDs and ultra disks. • Non-I/O optimized instances support only basic disks.	Generation I instance types are legacy shared instance types. They are still categorized based on the number of cores (1, 2, 4, 8, and 16 cores) and are not sensitive to instance families.

Instance family	Feature	Scenario
s3, generation I instance family	• Uses Intel® Xeon® E5-2420 processors that deliver a minimum clock speed of 1.9 GHz. • Uses the latest DDR3 memory. • Is an instance family in which instances can be I/O optimized or non-I/O optimized. • I/O optimized instances support both standard SSDs and ultra disks. • Non-I/O optimized instances support only basic disks.	Generation I instance types are legacy shared instance types. They are still categorized based on the number of cores (1, 2, 4, 8, and 16 cores) and are not sensitive to instance families.
t1, generation I instance family	• Uses Intel® Xeon® E5-2420 processors that deliver a minimum clock speed of 1.9 GHz. • Uses the latest DDR3 memory. • Is an instance family in which all instances are non-I/O optimized. • Supports only basic disks.	Generation I instance types are legacy shared instance types. They are still categorized based on the number of cores (1, 2, 4, 8, and 16 cores) and are not sensitive to instance families.

Instance families based on Hygon servers

Instance family	Feature	Scenario
ghg	ghg is a general purpose dedicated instance family that has the following features: • Offers a CPU-to-memory ratio of 1:4. • Uses the latest DDR4 memory. • Is an instance family in which all instances are I/O optimized by default. • Supports IPv6.	Small and medium-sized web servers, batch processing, advertising services, and distributed analysis
dhg	dhg is a big data instance family that has the following features: • Offers a CPU-to-memory ratio of 1:4. • Uses the latest DDR4 memory. • Is an instance family in which all instances are I/O optimized by default. • Supports local disks.	Industries such as Internet and finance that need to compute, store, and analyze big data

2.4.1.3. Instance types

An Elastic Compute Service (ECS) instance is the smallest unit that can provide compute capabilities and services for your business. Compute capabilities vary based on instance types.

An ECS instance type defines the basic properties of an ECS instance, such as CPU, clock speed, and memory. In addition to the instance type, you must also configure the Elastic Block Storage (EBS) devices, image, and network type when you create an ECS instance.

 **Note** The instance families and instance types described in this topic are for reference only. The physical server where an instance is hosted determines the instance type.

Instance types based on Intel x86 servers

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
	ecs.s6-c1m1.small	1	1	None	0.5	50	1	2
	ecs.s6-c1m2.small	1	2	None	0.5	50	1	2
	ecs.s6-c1m4.small	1	4	None	0.5	50	1	2
	ecs.s6-c1m8.small	1	8	None	0.5	50	1	2
	ecs.s6-c1m2.large	2	4	None	Baseline 0.5 and burstable up to 1.0	100	1	2
	ecs.s6-c1m4.large	2	8	None	Baseline 0.5 and burstable up to 1.0	100	1	2

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
s6, shared standard instance family	ecs.s6-c1m8.large	2	16	None	Baseline 0.5 and burstable up to 1.0	100	1	2
	ecs.s6-c1m2.xlarge	4	8	None	Baseline 0.8 and burstable up to 1.5	150	1	2
	ecs.s6-c1m4.xlarge	4	16	None	Baseline 0.8 and burstable up to 1.5	150	1	2
	ecs.s6-c1m8.xlarge	4	32	None	Baseline 0.8 and burstable up to 1.5	150	1	2
	ecs.s6-c1m2.2xlarge	8	16	None	Baseline 1.2 and burstable up to 3.0	300	1	2
	ecs.s6-c1m4.2xlarge	8	32	None	Baseline 1.2 and burstable up to 3.0	300	1	2

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
	ecs.s6-c1m8.2xlarge	8	64	None	Baseline 1.2 and burstable up to 3.0	300	1	2
	ecs.s6-c1m2.4xlarge	16	32	None	Baseline 2.0 and burstable up to 4.0	400	1	2
	ecs.s6-c1m4.4xlarge	16	64	None	Baseline 2.0 and burstable up to 4.0	400	1	2
	ecs.s6-c1m8.4xlarge	16	128	None	Baseline 2.0 and burstable up to 4.0	400	1	8
	ecs.s6-c1m2.8xlarge	32	64	None	Baseline 3.0 and burstable up to 5.0	500	1	2
	ecs.s6-c1m4.8xlarge	32	128	None	Baseline 3.0 and burstable up to 5.0	500	1	8

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
n4, shared general purpose instance family	ecs.n4.small	1	2.0	None	0.5	50	1	1
	ecs.n4.large	2	4.0	None	0.5	100	1	1
	ecs.n4.xlarge	4	8.0	None	0.8	150	1	2
	ecs.n4.2xlarge	8	16.0	None	1.2	300	1	2
	ecs.n4.4xlarge	16	32.0	None	2.5	400	1	2
	ecs.n4.8xlarge	32	64.0	None	5.0	500	1	2
mn4, shared balanced instance family	ecs.mn4.small	1	4.0	None	0.5	50	1	1
	ecs.mn4.large	2	8.0	None	0.5	100	1	1
	ecs.mn4.xlarge	4	16.0	None	0.8	150	1	2
	ecs.mn4.2xlarge	8	32.0	None	1.2	300	1	3
	ecs.mn4.4xlarge	16	64.0	None	2.5	400	1	8
	ecs.mn4.8xlarge	32	128.0	None	5.0	500	2	8
xn4, shared compact instance family	ecs.xn4.small	1	1.0	None	0.5	50	1	1

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
e4, shared memory optimized instance family	ecs.e4.small	1	8.0	None	0.5	50	1	1
	ecs.e4.large	2	16.0	None	0.5	100	1	1
	ecs.e4.xlarge	4	32.0	None	0.8	150	1	2
	ecs.e4.2xlarge	8	64.0	None	1.2	300	1	3
	ecs.e4.4xlarge	16	128.0	None	2.5	400	1	8
sn1ne, compute optimized instance family with enhanced network performance	ecs.sn1ne.large	2	4.0	None	1.0	300	2	2
	ecs.sn1ne.xlarge	4	8.0	None	1.5	500	2	3
	ecs.sn1ne.2xlarge	8	16.0	None	2.0	1,000	4	4
	ecs.sn1ne.3xlarge	12	24.0	None	2.5	1,300	4	6
	ecs.sn1ne.4xlarge	16	32.0	None	3.0	1,600	4	8
	ecs.sn1ne.6xlarge	24	48.0	None	4.5	2,000	6	4
	ecs.sn1ne.8xlarge	32	64.0	None	6.0	2,500	8	8
	ecs.g7.large	2	8	None	1.0	300	2	3
	ecs.g7.xlarge	4	16	None	1.0	500	4	4
	ecs.g7.2xlarge	8	32	None	2.0	800	8	4
	ecs.g7.3xlarge	12	48	None	2.0	900	8	8

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
g7, general purpose instance family	ecs.g7.xlarge	16	64	None	2.0	1,000	8	8
	ecs.g7.6xlarge	24	96	None	2.5	1,500	12	8
	ecs.g7.8xlarge	32	128	None	3.0	2,000	16	8
	ecs.g7.12xlarge	48	192	None	4.0	3,000	16	8
	ecs.g7.16xlarge	64	256	None	5.0	4,000	32	8
	ecs.g7.24xlarge	96	384	None	8.0	6,000	32	15
	ecs.g7.29xlarge	116	448	None	10.0	8,000	32	15
g6, general purpose instance family	ecs.g6.large	2	8.0	None	1.0	300	2	2
	ecs.g6.xlarge	4	16.0	None	1.5	500	4	3
	ecs.g6.2xlarge	8	32.0	None	2.5	800	8	4
	ecs.g6.3xlarge	12	48.0	None	4.0	900	8	6
	ecs.g6.4xlarge	16	64.0	None	5.0	1,000	8	8
	ecs.g6.6xlarge	24	96.0	None	7.5	1,500	12	8
	ecs.g6.8xlarge	32	128.0	None	10.0	2,000	16	8
	ecs.g5.large	2	8.0	None	1.0	300	2	2
	ecs.g5.xlarge	4	16.0	None	1.5	500	2	3

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
g5, general purpose instance family	ecs.g5.2xlarge	8	32.0	None	2.5	800	2	4
	ecs.g5.3xlarge	12	48.0	None	4.0	900	4	6
	ecs.g5.4xlarge	16	64.0	None	5.0	1,000	4	8
	ecs.g5.6xlarge	24	96.0	None	7.5	1,500	6	8
	ecs.g5.8xlarge	32	128.0	None	10.0	2,000	8	8
	ecs.g5.16xlarge	64	256.0	None	20.0	4,000	16	8
sn2ne, general purpose instance family with enhanced network performance	ecs.sn2ne.large	2	8.0	None	1.0	300	2	2
	ecs.sn2ne.xlarge	4	16.0	None	1.5	500	2	3
	ecs.sn2ne.2xlarge	8	32.0	None	2.0	1,000	4	4
	ecs.sn2ne.3xlarge	12	48.0	None	2.5	1,300	4	6
	ecs.sn2ne.4xlarge	16	64.0	None	3.0	1,600	4	8
	ecs.sn2ne.6xlarge	24	96.0	None	4.5	2,000	6	4
	ecs.sn2ne.8xlarge	32	128.0	None	6.0	2,500	8	8
	ecs.sn2ne.14xlarge	56	224.0	None	10.0	4,500	14	8

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
r6, memory optimized instance family	ecs.r6.large	2	16.0	None	1.0	300	2	2
	ecs.r6.xlarge	4	32.0	None	1.5	500	4	3
	ecs.r6.2xlarge	8	64.0	None	2.5	800	8	4
	ecs.r6.3xlarge	12	96.0	None	4.0	900	8	6
	ecs.r6.4xlarge	16	128.0	None	5.0	1,000	8	8
	ecs.r6.6xlarge	24	192.0	None	7.5	1,500	12	8
	ecs.r6.8xlarge	32	256.0	None	10.0	2,000	16	8
r5, memory optimized instance family	ecs.r5.large	2	16.0	None	1.0	300	2	2
	ecs.r5.xlarge	4	32.0	None	1.5	500	2	3
	ecs.r5.2xlarge	8	64.0	None	2.5	800	2	4
	ecs.r5.3xlarge	12	96.0	None	4.0	900	4	6
	ecs.r5.4xlarge	16	128.0	None	5.0	1,000	4	8
	ecs.r5.6xlarge	24	192.0	None	7.5	1,500	6	8
	ecs.r5.8xlarge	32	256.0	None	10.0	2,000	8	8
	ecs.r5.16xlarge	64	512.0	None	20.0	4,000	16	8

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
se1ne, memory optimized instance family with enhanced network performance	ecs.se1ne.large	2	16.0	None	1.0	300	2	2
	ecs.se1ne.xlarge	4	32.0	None	1.5	500	2	3
	ecs.se1ne.2xlarge	8	64.0	None	2.0	1,000	4	4
	ecs.se1ne.3xlarge	12	96.0	None	2.5	1,300	4	6
	ecs.se1ne.4xlarge	16	128.0	None	3.0	1,600	4	8
	ecs.se1ne.6xlarge	24	192.0	None	4.5	2,000	6	4
	ecs.se1ne.8xlarge	32	256.0	None	6.0	2,500	8	8
	ecs.se1ne.14xlarge	56	480.0	None	10.0	4,500	14	8
se1, memory optimized instance family	ecs.se1.large	2	16.0	None	0.5	100	1	2
	ecs.se1.xlarge	4	32.0	None	0.8	200	1	3
	ecs.se1.2xlarge	8	64.0	None	1.5	400	1	4
	ecs.se1.4xlarge	16	128.0	None	3.0	500	2	8
	ecs.se1.8xlarge	32	256.0	None	6.0	800	3	8
	ecs.se1.14xlarge	56	480.0	None	10.0	1,200	4	8

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
ebmg5s, general purpose ECS Bare Metal Instance family with enhanced network performance	ecs.ebmg5s.24xlarge	96	384.0	None	30.0	4,500	8	32
ebmg5, general purpose ECS Bare Metal Instance family	ecs.ebmg5.24xlarge	96	384.0	None	10.0	4,000	8	32
i2, instance family with local SSDs	ecs.i2.xlarge	4	32.0	1 × 894	1.0	500	2	3
	ecs.i2.2xlarge	8	64.0	1 × 1,788	2.0	1,000	2	4
	ecs.i2.4xlarge	16	128.0	2 × 1,788	3.0	1,500	4	8
	ecs.i2.8xlarge	32	256.0	4 × 1,788	6.0	2,000	8	8
	ecs.i2.16xlarge	64	512.0	8 × 1,788	10.0	4,000	16	8
	ecs.d1.2xlarge	8	32.0	4 × 5,500	3.0	300	1	4
	ecs.d1.3xlarge	12	48.0	6 × 5,500	4.0	400	1	6
	ecs.d1.4xlarge	16	64.0	8 × 5,500	6.0	600	2	8

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
d1, big data instance family	ecs.d1.6xlarge	24	96.0	12 × 5,500	8.0	800	2	8
	ecs.d1-c8d3.8xlarge	32	128.0	12 × 5,500	10.0	1,000	4	8
	ecs.d1.8xlarge	32	128.0	16 × 5,500	10.0	1,000	4	8
	ecs.d1-c14d3.14xlarge	56	160.0	12 × 5,500	17.0	1,800	6	8
	ecs.d1.14xlarge	56	224.0	28 × 5,500	17.0	1,800	6	8
d2, big data instance family	ecs.d2-zyy-d0.4xlarge	16	64.0	None	3.0	300	2	8
	ecs.d2-zyy-d0.6xlarge	24	96.0	None	4.0	400	2	8
	ecs.d2-zyy.4xlarge	16	64.0	6 × 7,500	3.0	300	4	8
	ecs.d2-zyy.6xlarge	24	96.0	12 × 7,500	4.0	400	4	8
	ecs.d2-zyy.7xlarge	56	224.0	12 × 7,300	17.0	1,800	6	8
	ecs.d2-zyy.8xlarge	32	160.0	12 × 7,300	17.0	1,800	6	8
	ecs.d2-zyy.22xlarge	88	352.0	12 × 7,300	17.0	1,800	6	8

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
	ecs.d2-zyy-m40.8xlarge	32	128.0	12 × 7,500	6.0	600	4	8
	ecs.d2-gab.4xlarge	16	64.0	6 × 1,150	3.0	300	4	8
	ecs.d2-gab.8xlarge	32	128.0	12 × 1,150	6.0	600	4	8
sccg5ib, general purpose SCC instance family	ecs.sccg5ib.24xlarge	96	384.0	None	10.0	4,500	8	32
scch5ib, SCC instance family with high clock speeds	ecs.scch5ib.16xlarge	64	192.0	None	10.0	4,500	8	32
c6, compute optimized instance family	ecs.c6.large	2	4.0	None	1.0	300	2	2
	ecs.c6.xlarge	4	8.0	None	1.5	500	4	3
	ecs.c6.2xlarge	8	16.0	None	2.5	800	8	4
	ecs.c6.3xlarge	12	24.0	None	4.0	900	8	6
	ecs.c6.4xlarge	16	32.0	None	5.0	1,000	8	8
	ecs.c6.6xlarge	24	48.0	None	7.5	1,500	12	8

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
	ecs.c6.8xlarge	32	64.0	None	10.0	2,000	16	8
sn1, compute optimized instance family	ecs.sn1.medium	2	4.0	None	0.5	100	1	2
	ecs.sn1.large	4	8.0	None	0.8	200	1	3
	ecs.sn1.xlarge	8	16.0	None	1.5	400	1	4
	ecs.sn1.3xlarge	16	32.0	None	3.0	500	2	8
	ecs.sn1.7xlarge	32	64.0	None	6.0	800	3	8
sn2, general purpose instance family	ecs.sn2.medium	2	8.0	None	0.5	100	1	2
	ecs.sn2.large	4	16.0	None	0.8	200	1	3
	ecs.sn2.xlarge	8	32.0	None	1.5	400	1	4
	ecs.sn2.3xlarge	16	64.0	None	3.0	500	2	8
	ecs.sn2.7xlarge	32	128.0	None	6.0	800	3	8
	ecs.sn2.13xlarge	56	224.0	None	10.0	1,200	4	8

The following table describes FPGA-accelerated instance families.

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)	FPGA
f1, FPGA-accelerated compute optimized instance family	ecs.f1-c8f1.2xlarge	8	60.0	None	3.0	400	4	2	Intel ARRIA 10 GX 1150
	ecs.f1-c8f1.4xlarge	16	120.0	None	5.0	1,000	4	2	2 × Intel ARRIA 10 GX 1150
	ecs.f1-c28f1.7xlarge	28	112.0	None	5.0	2,000	8	2	Intel ARRIA 10 GX 1150
	ecs.f1-c28f1.14xlarge	56	224.0	None	10.0	2,000	14	2	2 × Intel ARRIA 10 GX 1150
f3, FPGA-accelerated compute optimized instance family	ecs.f3-c16f1.4xlarge	16	64.0	None	5.0	1,000	4	8	1 × Xilinx VU9P
	ecs.f3-c16f1.8xlarge	32	128.0	None	10.0	2,000	8	8	2 × Xilinx VU9P
	ecs.f3-c16f1.16xlarge	64	256.0	None	20.0	2,000	16	8	4 × Xilinx VU9P

The following table describes GPU-accelerated instance families.

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)	GPU
gn5, GPU-accelerated compute optimized instance family	ecs.gn5-c4g1.xlarge	4	30.0	440	3.0	300	1	3	1 × NVIDIA P100
	ecs.gn5-c8g1.2xlarge	8	60.0	440	3.0	400	1	4	1 × NVIDIA P100
	ecs.gn5-c4g1.2xlarge	8	60.0	880	5.0	1,000	2	4	2 × NVIDIA P100
	ecs.gn5-c8g1.4xlarge	16	120.0	880	5.0	1,000	4	8	2 × NVIDIA P100
	ecs.gn5-c28g1.7xlarge	28	112.0	440	5.0	1,000	8	8	1 × NVIDIA P100
	ecs.gn5-c8g1.8xlarge	32	240.0	1,760	10.0	2,000	8	8	4 × NVIDIA P100
	ecs.gn5-c28g1.14xlarge	56	224.0	880	10.0	2,000	14	8	2 × NVIDIA P100
	ecs.gn5-c8g1.14xlarge	54	480.0	3,520	25.0	4,000	14	8	8 × NVIDIA P100

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)	GPU
gn4, GPU-accelerated compute optimized instance family	ecs.gn4-c4g1.xlarge	4	30.0	None	3.0	300	1	3	1 × NVIDIA M40
	ecs.gn4-c8g1.2xlarge	8	30.0	None	3.0	400	1	4	1 × NVIDIA M40
	ecs.gn4.8xlarge	32	48.0	None	6.0	800	3	8	1 × NVIDIA M40
	ecs.gn4-c4g1.2xlarge	8	60.0	None	5.0	500	1	4	2 × NVIDIA M40
	ecs.gn4-c8g1.4xlarge	16	60.0	None	5.0	500	1	8	2 × NVIDIA M40
	ecs.gn4.14xlarge	56	96.0	None	10.0	1,200	4	8	2 × NVIDIA M40
ga1, GPU-accelerated visualization and compute optimized instance family	ecs.ga1.xlarge	4	10.0	1 × 87	1.0	200	1	3	0.25 × AMD S7150
	ecs.ga1.2xlarge	8	20.0	1 × 175	1.5	300	1	4	0.5 × AMD S7150
	ecs.ga1.4xlarge	16	40.0	1 × 350	3.0	500	2	8	1 × AMD S7150
	ecs.ga1.8xlarge	32	80.0	1 × 700	6.0	800	3	8	2 × AMD S7150
	ecs.ga1.14xlarge	56	160.0	1 × 1,400	10.0	1,200	4	8	4 × AMD S7150

family									
Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)	GPU
gn5i, GPU-accelerated compute optimized instance family	ecs.gn5i-c2g1.large	2	8.0	None	1.0	100	2	2	1 × NVIDIA P4
	ecs.gn5i-c4g1.xlarge	4	16.0	None	1.5	200	2	3	1 × NVIDIA P4
	ecs.gn5i-c8g1.2xlarge	8	32.0	None	2.0	400	4	4	1 × NVIDIA P4
	ecs.gn5i-c16g1.4xlarge	16	64.0	None	3.0	800	4	8	1 × NVIDIA P4
	ecs.gn5i-c16g1.8xlarge	32	128.0	None	6.0	1,200	8	8	2 × NVIDIA P4
	ecs.gn5i-c24g1.12xlarge	48	192.0	None	10.0	2,000	8	8	2 × NVIDIA P4
	ecs.gn5i-c28g1.14xlarge	56	224.0	None	10.0	2,000	14	8	2 × NVIDIA P4

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)	GPU
gn5e, GPU-accelerated compute optimized instance family	ecs.gn5e-c11g1.3xlarge	10	58.0	None	2.0	150	1	6	1 × NVIDIA P4
	ecs.gn5e-c11g1.5xlarge	22	116.0	None	4.0	300	1	8	2 × NVIDIA P4
	ecs.gn5e-c11g1.11xlarge	44	232.0	None	6.0	600	2	8	4 × NVIDIA P4
	ecs.gn5e-c11g1.22xlarge	88	464.0	None	10.0	1,200	4	8	8 × NVIDIA P4
	ecs.gn6i-c4g1.xlarge	4	15	None	4.0	500	2	2	1 × T4
	ecs.gn6i-c8g1.2xlarge	8	31	None	5.0	800	2	2	1 × T4
	ecs.gn6i-c10g1.2xlarge	10	42.0	None	5.0	800	2	4	1 × T4
	ecs.gn6i-c10g1.5xlarge	20	84.0	None	8.0	1,000	4	6	2 × T4
	ecs.gn6i-c10g1.10xlarge	40	168.0	None	15.0	2,000	8	8	4 × T4

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)	GPU
gn6i, GPU-accelerated compute optimized instance family	ecs.gn6i-c10g1.20xlarge	80	336.0	None	30.0	4,000	16	8	8 × T4
	ecs.gn6i-c14g1.3xlarge	14	56.0	None	5.0	1,000	4	6	1 × T4
	ecs.gn6i-c14g1.7xlarge	28	112.0	None	10.0	2,000	8	8	2 × T4
	ecs.gn6i-c14g1.14xlarge	56	224.0	None	20.0	4,000	12	8	4 × T4
	ecs.gn6i-c16g1.4xlarge	16	62	None	6.0	1,000	4	3	1 × T4
	ecs.gn6i-c20g1.5xlarge	20	80.0	None	10.0	1,500	4	6	1 × T4
	ecs.gn6i-c20g1.10xlarge	40	160.0	None	20.0	3,000	8	8	2 × T4
	ecs.gn6i-c24g1.6xlarge	24	93	None	7.5	1,200	6	4	1 × T4

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)	GPU
	ecs.gn6i-c24g1.12xlarge	48	186	None	15.0	2,400	12	6	2 × T4
	ecs.gn6i-c24g1.24xlarge	96	372	None	30.0	4,800	24	8	4 × T4
	ecs.gn6i-c32g1.8xlarge	32	124	None	10.0	2,500	8	3	1 × T4
	ecs.gn6i-c48g1.12xlarge	48	186	None	15.0	4,500	12	6	1 × T4
	ecs.gn6i-c72g1.18xlarge	72	279	None	21.5	4,500	18	8	1 × T4
gn6v, GPU-accelerated compute optimized instance family	ecs.gn6v-c8g1.2xlarge	8	32.0	None	2.5	800	4	4	1 × NVIDIA V100
	ecs.gn6v-c8g1.8xlarge	32	128.0	None	10.0	2,000	8	8	4 × NVIDIA V100
	ecs.gn6v-c8g1.16xlarge	64	256.0	None	20.0	2,500	16	8	8 × NVIDIA V100

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)	GPU
vgn6i, lightweight GPU-accelerated compute optimized instance family	ecs.vgn6i-m1.xlarge	4	20.5	None	1.5	300	4	3	1/16 × T4
	ecs.vgn6i-m2.large	2	11	None	1	300	2	2	1/8 × T4
	ecs.vgn6i-m2.xlarge	4	20.5	None	2	400	4	3	1/8 × T4
	ecs.vgn6i-m4.xlarge	4	23.0	None	2	500	4	3	1/4 × T4
	ecs.vgn6i-m4.3xlarge	12	60.0	None	4	750	8	3	1/4 × T4
	ecs.vgn6i-m8.2xlarge	10	46.0	None	4	800	8	4	1/2 × T4
	ecs.vgn6i-m8.6xlarge	24	120.0	None	8	1,500	16	3	1/2 × T4

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)	GPU
sccgn6p, GPU-accelerated compute optimized SCC instance family	ecs.sccgn6p.24xlarge	96	768.0	None	30.0	4,500	8	32	8 × NVIDIA V100

The following table describes shared instance families that support IPv6.

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	IPv6 support	NIC queues	ENIs (including one primary ENI)
n4v2, shared general purpose instance family	ecs.n4v2.small	1	2.0	None	0.5	50	Yes	1	1
	ecs.n4v2.large	2	4.0	None	0.5	100	Yes	1	1
	ecs.n4v2.xlarge	4	8.0	None	0.8	150	Yes	1	2
	ecs.n4v2.2xlarge	8	16.0	None	1.2	300	Yes	1	2
	ecs.n4v2.4xlarge	16	32.0	None	2.5	400	Yes	1	2
	ecs.n4v2.8xlarge	32	64.0	None	5.0	500	Yes	1	2

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	IPv6 support	NIC queues	ENIs (including one primary ENI)
mn4v2, shared balanced instance family	ecs.mn4v2.small	1	4.0	None	0.5	50	Yes	1	1
	ecs.mn4v2.large	2	8.0	None	0.5	100	Yes	1	1
	ecs.mn4v2.xlarge	4	16.0	None	0.8	150	Yes	1	2
	ecs.mn4v2.2xlarge	8	32.0	None	1.2	300	Yes	1	2
	ecs.mn4v2.4xlarge	16	64.0	None	2.5	400	Yes	1	2
	ecs.mn4v2.8xlarge	32	128.0	None	5.0	500	Yes	2	8
xn4v2, shared compact instance family	ecs.xn4v2.small	1	1.0	None	0.5	50	Yes	1	1
e4v2, shared memory optimized instance family	ecs.e4v2.small	1	8.0	None	0.5	50	Yes	1	1
	ecs.e4v2.large	2	16.0	None	0.5	100	Yes	1	1
	ecs.e4v2.xlarge	4	32.0	None	0.8	150	Yes	1	2
	ecs.e4v2.2xlarge	8	64.0	None	1.2	300	Yes	1	3
	ecs.e4v2.4xlarge	16	128.0	None	2.5	400	Yes	1	8

The following table describes burstable instance families.

Instance family	Instance type	vCPUs	Memory (GiB)	Baseline CPU computing performance	CPU credits per hour	Max CPU credit balance	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
	ecs.t5-lc2m1.nano	1	0.5	20%	12	288	None	0.1	40	1	1
	ecs.t5-lc1m1.small	1	1.0	20%	12	288	None	0.2	60	1	1
	ecs.t5-lc1m2.small	1	2.0	20%	12	288	None	0.2	60	1	1
	ecs.t5-lc1m2.large	2	4.0	20%	24	576	None	0.4	100	1	1
	ecs.t5-lc1m4.large	2	8.0	20%	24	576	None	0.4	100	1	1
	ecs.t5-c1m1.large	2	2.0	25%	30	720	None	0.5	100	1	1
	ecs.t5-c1m2.large	2	4.0	25%	30	720	None	0.5	100	1	1

Instance family	Instance type	vCPUs	Memory (GiB)	Baseline CPU computing performance	CPU credits per hour	Max CPU credit balance	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
t5, burstable instance family	ecs.t5-c1m4.large	2	8.0	25%	30	720	None	0.5	100	1	1
	ecs.t5-c1m1.xlarge	4	4.0	25%	60	1,440	None	0.8	200	1	2
	ecs.t5-c1m2.xlarge	4	8.0	25%	60	1,440	None	0.8	200	1	2
	ecs.t5-c1m4.xlarge	4	16.0	25%	60	1,440	None	0.8	200	1	2
	ecs.t5-c1m1.2xlarge	8	8.0	25%	120	2,880	None	1.2	400	1	2
	ecs.t5-c1m2.2xlarge	8	16.0	25%	120	2,880	None	1.2	400	1	2
	ecs.t5-c1m4.2xlarge	8	32.0	25%	120	2,880	None	1.2	400	1	2

Instance family	Instance type	vCPUs	Memory (GiB)	Baseline CPU computing performance	CPU credits per hour	Max CPU credit balance	Local storage (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
	ecs.t5-c1m1.4xlarge	16	16.0	25%	240	5,760	None	1.2	600	1	2
	ecs.t5-c1m2.4xlarge	16	32.0	25%	240	5,760	None	1.2	600	1	2

The following table describes the ecs.anyshare custom instance type.

Instance type	vCPUs (x)	Memory (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)	Disk bandwidth (Gbit/s)
Custom instance type ecs.anyshare	$0 < x \leq 2$	1 to 16	0.5	100	1	2	0.5
	$2 < x \leq 4$	2 to 32	0.8	$100 + (x - 2)/0.2$	1	2	0.8
	$4 < x \leq 8$	4 to 64	$0.8 + (x - 5)/4$	$200 + (x - 4)/0.2$	1	3	$0.8 + (x - 5)/4$
	$8 < x \leq 12$	8 to 96	$1.5 + (x - 8)/8$	$400 + (x - 8)/0.8$	2	3	$1.5 + (x - 8)/4$
	$12 < x \leq 16$	12 to 128	$2 + (x - 12)/4$	$450 + (x - 12)/0.8$	3	4	1
	$16 < x \leq 24$	16 to 196	$3 + (x - 16)/8$	$500 + (x - 16)/0.8$	3	5	$1 + (x - 16)/8$
	$24 < x \leq 32$	24 to 256	$4 + (x - 24)/8$	$600 + (x - 24)/0.4$	4	6	$2 + (x - 24)/8$

Instance type	vCPUs (x)	Memory (GiB)	Bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)	Disk bandwidth (Gbit/s)
	$x > 32$	32 to 352	$\min(5 + (x - 32)/8, 10)$	$\min(800 + (x - 32)/0.8, 1,200)$	4	8	$\min(3 + (x - 32)/8, 8)$

The following instance types are applicable only in environments that are upgraded from Apsara Stack V2 to V3.

Instance family	Instance type	vCPUs	Memory (GiB)
n1	ecs.n1.tiny	1	1.0
	ecs.n1.small	1	2.0
	ecs.n1.medium	2	4.0
	ecs.n1.large	4	8.0
	ecs.n1.xlarge	8	16.0
	ecs.n1.3xlarge	16	32.0
	ecs.n1.7xlarge	32	64.0
n2	ecs.n2.small	1	4.0
	ecs.n2.medium	2	8.0
	ecs.n2.large	4	16.0
	ecs.n2.xlarge	8	32.0
	ecs.n2.3xlarge	16	64.0
	ecs.n2.7xlarge	32	128.0
e3	ecs.e3.small	1	8.0
	ecs.e3.medium	2	16.0
	ecs.e3.large	4	32.0
	ecs.e3.xlarge	8	64.0
	ecs.e3.3xlarge	16	128.0
	ecs.c1.small	8	8.0

c1 Instance family	Instance type	vCPUs	Memory (GiB)
	ecs.c1.large	8	16.0
c2	ecs.c2.medium	16	16.0
	ecs.c2.large	16	32.0
	ecs.c2.xlarge	16	64.0
m1	ecs.m1.medium	4	16.0
	ecs.m1.xlarge	8	32.0
m2	ecs.m2.medium	4	32.0
s1	ecs.s1.small	1	2.0
	ecs.s1.medium	1	4.0
	ecs.s1.large	1	8.0
s2	ecs.s2.small	2	2.0
	ecs.s2.large	2	4.0
	ecs.s2.xlarge	2	8.0
	ecs.s2.2xlarge	2	16.0
s3	ecs.s3.medium	4	4.0
	ecs.s3.large	4	8.0
t1	ecs.t1.small	1	1.0

Instance types based on Hygon servers

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Baseline bandwidth (bidirectional), Gbit/s	Burstable bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
	ecs.ghg.large	2	8	None	0.5	1.5	100	1	2
	ecs.ghg.xlarge	4	16	None	0.8	1.5	200	1	2

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Baseline bandwidth (bidirectional), Gbit/s	Burstable bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
ghg	ecs.ghg.2xlarge	8	32	None	1.5	2	400	1	3
	ecs.ghg.4xlarge	16	64	None	3	None	500	3	4
	ecs.ghg.8xlarge	32	128	None	5	None	800	4	6
	ecs.ghg.16xlarge	64	256	None	7	None	1,000	4	8
	ecs.ghg.24xlarge	96	384	None	8	None	1,200	4	8
	ecs.ghg.30xlarge	120	480	None	10	None	1,200	4	8
dhg	ecs.dhg.2xlarge	8	32	4 × 7,300	3	None	300	1	4
	ecs.dhg.4xlarge	16	64	4 × 7,300	6	None	600	2	8
	ecs.dhg.6xlarge	24	96	6 × 7,300	8	None	800	2	8
	ecs.dhg.8xlarge	32	128	6 × 7,300	10	None	1,000	4	8
	ecs.dhg.12xlarge	48	196	8 × 7,300	15	None	1,500	4	8

Instance family	Instance type	vCPUs	Memory (GiB)	Local storage (GiB)	Baseline bandwidth (bidirectional), Gbit/s	Burstable bandwidth (bidirectional), Gbit/s	Packet forwarding rate (bidirectional), Kpps	NIC queues	ENIs (including one primary ENI)
	ecs.dh.g.16xlarge	64	256	8 × 7,300	15	None	1,500	4	8
	ecs.dh.g.24xlarge	96	384	12 × 7,300	17	None	1,800	6	8
	ecs.dh.g.30xlarge	120	480	12 × 7,300	17	None	1,800	6	8

2.4.1.4. UserData

UserData allows you to customize the startup behavior of instances and import data to ECS instances. It is the basis for ECS instance customization.

UserData is implemented through different types of scripts. Before UserData is implemented on an instance, all ECS instances will have the same initial environment and configurations when started for the first time. After enterprises or individuals enter valid UserData information based on their scenarios and needs, required ECS instances are provided after the first startup.

Methods

- **UserData-Scripts:** are applicable to users who need to initialize instances by executing the shell scripts. The UserData-Scripts begin with `#!/bin/sh`. A review of user data shows that most users input UserData by running UserData-Scripts. UserData-Scripts are also suitable for complicated deployment scenarios.
- **Cloud-Config:** is a special script supported by cloud-init. It packs frequently-used personalized configurations into YAML files, which enable you to complete the frequently-used configurations more conveniently. The script starts with `#Cloud-config` in the first line and is followed by an array containing `ssh_authorized_keys`, `hostname`, `write_files`, and `manage_etc_hosts`.

Scenarios

- SSH authentication
- Software source updates and configuration
- DNS configuration
- Application installation and configuration

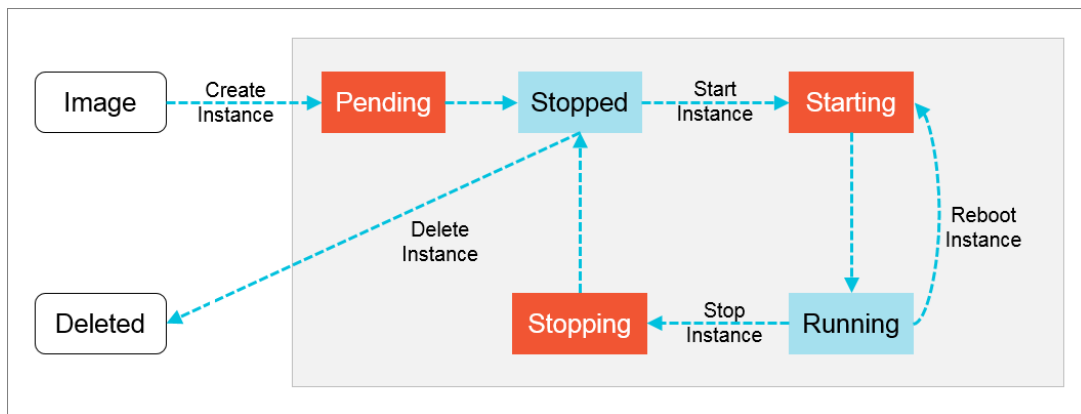
2.4.1.5. Instance lifecycle

The lifecycle of an ECS instance begins when the instance is created and ends when the instance is released. This topic describes the instance states in the ECS console as well as state attributes and their corresponding instance states in API responses.

The following table describes the instance states in the ECS console and their corresponding instance states in API responses.

State	State attribute	Description	State in an API response
Instance being created	Intermediate	The instance is being created and waiting to start. If an instance remains in this state for an extended period of time, an exception has occurred.	Pending
Starting	Intermediate	When you start or restart an instance by using the ECS console or by calling an API operation, the instance enters this state before it enters the Running state. If an instance remains in the Starting state for an extended period of time, an exception has occurred.	Starting
Running	Stable	While an instance is in the Running state, the instance can function normally and can accommodate your business needs.	Running
Stopping	Intermediate	When you stop an instance by using the ECS console or by calling an API operation, the instance enters this state before it enters the Stopped state. If an instance remains in the Stopping state for an extended period of time, an exception has occurred.	Stopping
Stopped	Stable	An instance enters this state when it is stopped. Instances in the Stopped state cannot provide external services.	Stopped
Reinitializing	Intermediate	When you re-initialize the system disk or a data disk of an instance by using the ECS console or calling an API operation, the instance enters this state before it enters the Running state. If an instance remains in the Reinitializing state for an extended period of time, an exception has occurred.	Stopped
Changing system disk	Intermediate	When you replace the system disk of an instance by using the ECS console or by calling an API operation, the instance enters this state before it enters the Running state. If an instance remains in the Changing system disk state for an extended period of time, an exception has occurred.	Stopped

The following figure shows the transitions between instance states in API responses.



2.4.1.6. EBM Instances

ECS Bare Metal (EBM) Instance is a new computing service that combines the elasticity of virtual machines with the performance and features of physical machines. EBM Instances are designed based on the state-of-the-art virtualization technology developed by Alibaba Cloud.

The virtualization used by EBM Instances is optimized to support common ECS instances and nested virtualization, maintaining elastic performance with the user experience of physical servers.

Benefits

EBM Instances provides the following benefits through technological innovation:

- **Exclusive computing resources**

As a cloud-based elastic computing service, EBM Instances surpass the performance and isolation of physical servers, enabling you to exclusively occupy computing resources without virtualization performance overhead or feature loss. EBM Instances support ultrahigh-frequency instances and can contain 8, 16, 32, or 96 CPU cores. An EBM Instance with eight CPU cores supports ultrahigh frequency processing from 3.7 to 4.1 GHz, providing better performance and response for gaming and finance businesses than peer services.

- **Encrypted computing**

For security, EBM Instances use a chip-level trusted execution environment (Intel® SGX) in addition to physical server isolation to ensure that encrypted data can only be computed within a secure and trusted environment. This chip-level hardware security protection provides a safe box for the data of cloud users and allows users to control all data encryption and key protection processes.

- **Any Stack on Alibaba Cloud**

An EBM Instance combines the performance strengths and complete features of physical machines and the ease-of-use and cost-effectiveness of cloud servers. It can effectively meet the demands of high-performance computing and help you build new hybrid clouds. Thanks to the flexibility, elasticity, and all the other strengths inherited from both virtual and physical machines, EBM Instances are endowed with re-virtualization ability. Offline private clouds can be seamlessly migrated to Alibaba Cloud without the performance overhead that may arise from nested virtualization, giving you a new approach for moving businesses onto the cloud.

- **Heterogeneous instruction set processor support**

The virtualization 2.0 technology used by EBM Instances is developed independently by Alibaba Cloud and supports ARM and other instruction set processors at no additional cost.

Configuration features

The following table lists the configuration features of EBM Instances.

Features

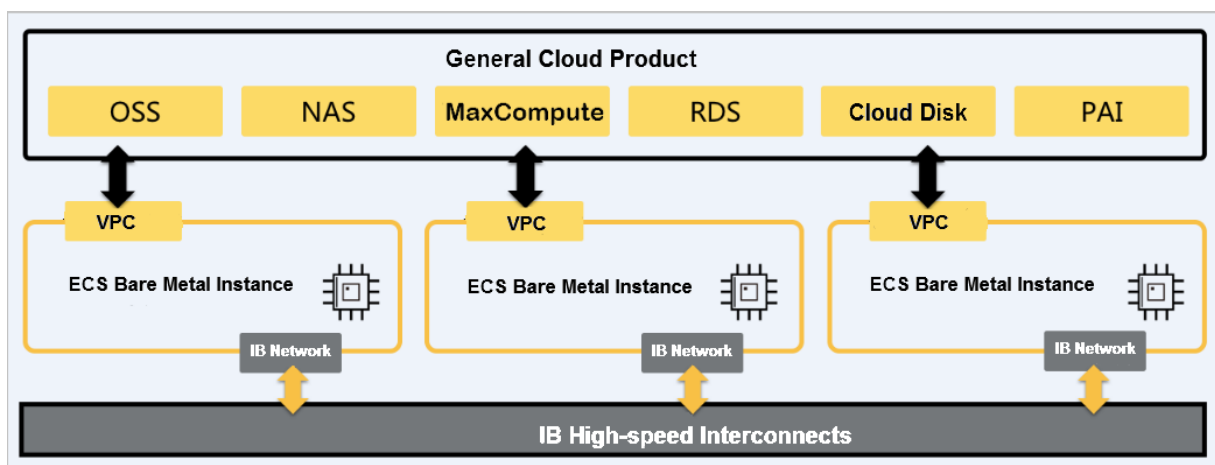
Item	Description
CPU configuration	Only supports the ebmg5 general-purpose EBM Instance type family.
Memory configuration	Supports expansion from 32 GiB to 384 GiB as needed. The ratio of CPU to memory is 1:2 or 1:4 to provide better computing performance.
Storage configuration	Supports startup from virtual machine images or cloud disks, achieving delivery in several seconds.
Network configuration	Supports Virtual Private Clouds (VPCs), and interoperability with ECS, GPU, and other cloud services. Delivers performance and stability comparable to physical machine networks.
Image configuration	Supports ECS images.
Security configuration	Maintains the same security policies and flexibility as existing ECS instances.

2.4.1.7. Super Computing Clusters

Super Computing Clusters (SCCs) provide computing cluster services with ultimate computing performance and parallel efficiency by integrating CPUs and heterogeneous accelerators such as GPUs that are interconnected through the high-speed InfiniBand (IB) network. SCCs are suited for scenarios such as high-performance computing, artificial intelligence, machine learning, scientific and engineering computing, data analysis, and audio and video processing.

SCC architecture

The following figure shows the SCC architecture.



SCCs are based on ECS Bare Metal Instances. By integrating the high-speed interconnects of InfiniBand technology and heterogeneous accelerators such as GPUs, SCCs have the following features:

- SCCs have all the benefits of ECS Bare Metal Instances. The underlying architecture allows you to use exclusive cloud servers or physical servers to create a secure and controllable underlying environment where you can configure security groups and VPCs for your SCC instances to implement traffic control.
- SCCs adopt InfiniBand, a conversion cable technology that supports multiple concurrent connections. InfiniBand is the next-generation I/O standard for compute server platforms and features high scalability, high bandwidth, and low latency. InfiniBand is ideal for establishing communication between servers such as replication servers and distributed servers, between servers and storage devices such as SAN and direct-attached storage, and between servers and networks such as LANs, WANs, and the Internet. The InfiniBand architecture is commonly used in high-performance computing and provides higher bandwidth, lower latency, and more reliable connections than the Ethernet architecture.

You can build your High Performance Computing (HPC) system based on SCCs.

Scenarios

Apsara Stack SCCs offer mature and flexible industry solutions and are suited for the following scenarios:

Scenario	Description
Large-scale AI computing	SCC-based HPC provides the computing capabilities that are required by large-scale AI computing for quick handling of problems associated with data and models. HPC and large-scale AI computing scenarios have similar requirements during planning, designing, and deployment phases.
Hybrid cloud for HPC	Both Apsara Stack and Alibaba Cloud public cloud have the same architecture for HPC, bringing a consistent hybrid cloud experience. You can use the E-HPC service provided by the Alibaba Cloud public cloud to migrate computing workloads from Apsara Stack to the public cloud. Before migrating your workload, you must activate resources on the public cloud and schedule tasks based on the E-HPC scheduler to build a hybrid cloud for HPC.
Supercomputing center in the cloud	Supercomputing centers are the earliest and most mature IT service developed in China. Traditional supercomputing centers integrate technologies from multiple vendors, which requires lengthy and complex technical solution consultation, equipment selection, and verification. The construction cycle is long, and operations and maintenance are complex. Apsara Stack supercomputing solutions can be used to build cloud supercomputing centers that provide large-scale supercomputing services in an agile and flexible manner. The all-in-one cloud for supercomputing and automated O&M reduce the complexity of operations and maintenance.

Scenario	Description
Industry verticals	The application of HPC in traditional industries is mature and continues to evolve. Apsara Stack SCCs are suited for all the industries where traditional supercomputing is applied. The industries include the following: • Petrochemical: seismic data processing and reservoir simulation • Finance: financial derivative analysis, actuarial analysis, asset and liability management, and investment risk analysis • Industrial manufacturing: collision analysis, failure analysis, and thermodynamic analysis • Life science: drug discovery, protein folding, DNA sequencing, and medical imaging • Media and entertainment: video post-production and animation rendering • Government and higher education: weather forecast, high-energy physics, and geophysics

2.4.2. Block storage

2.4.2.1. Overview

This topic describes the different types of Block Storage devices, including Elastic Block Storage (EBS) services based on a distributed storage architecture and local storage services based on the local hard disks of physical servers that host ECS instances.

Definitions of EBS and local storage:

- **Elastic Block Storage** provides ECS instances with block-level storage that features low latency, high performance, durability, and high reliability. A triplicate distributed mechanism is used to ensure data durability. EBS devices can be created, released, and resized at any time.

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.

- **Local storage**, also known as local disks, are temporary disks attached to physical machines that host ECS instances. Local storage is designed for business scenarios that require high storage I/O performance. It provides block-level data access capabilities for ECS instances and features low latency, high random IOPS, and high throughput.

Differences between Apsara Stack storage services

Apsara Stack provides the following data storage services: Block Storage, Object Storage Service (OSS), and Apsara File Storage NAS. The following table describes the differences among these services.

Comparison between data storage services

Data storage service	Feature	Scenario

Data storage service	Feature	Scenario
Block Storage	A high-performance and low-latency block-level storage service provided by Alibaba Cloud for ECS instances. It supports random read and write operations. You can format a Block Storage device and create file systems on it in the same way as you do with a physical disk.	Block Storage can meet the data storage requirements of most business scenarios.
OSS	A huge storage space designed to store unstructured data such as images, audios, and videos on the Internet. You can access data stored in OSS anytime and anywhere by calling API operations.	OSS is applicable to scenarios such as website construction, separation of dynamic and static resources, and CDN acceleration of domain name access.
Apsara File Storage NAS	A storage space designed to store large volumes of unstructured data that can be accessed based on standard file access protocols such as the Network File System (NFS) protocol for Linux and the Common Internet File System (CIFS) protocol for Windows. You can set permissions to allow different clients to concurrently access the same file.	Apsara File Storage NAS is applicable to scenarios such as file sharing across departments in an enterprise, non-linear editing in radio and television industries, high-performance computing, and Docker containers.

2.4.2.2. Elastic block storage

2.4.2.2.1. Overview

Elastic block storage can be divided into the following types based on whether it can be attached to multiple ECS instances.

- **Cloud disks:** A cloud disk can be attached to a single ECS instance that resides in the same zone and region.
- **Shared block storage:** A shared block storage can be attached to up to four ECS instances that belong to the same zone and region.

2.4.2.2.2. Cloud disks

Cloud disks are block-level storage devices provided by Apsara Stack for ECS instances. Cloud disks can be classified based on performance or purpose.

Performance-based classification

Cloud disks are divided by performance into ultra disks and standard SSDs.

- Ultra disks are ideal for medium I/O load scenarios and provide a random IOPS performance of up to 3,000 for ECS instances.
- Standard SSDs are ideal for I/O-intensive scenarios and provide stable and high random IOPS performance.

The following table provides a comparison of the performance of standard SSDs and ultra disks.

Item	Standard SSD	Ultra disk
Maximum capacity of a single disk	32,768 GiB	32,768 GiB
Maximum IOPS	25,000	5,000
Maximum throughput	300 MB/s	140 MB/s
Formula for calculating the IOPS per disk	$\min\{1800 + 30 \times \text{Capacity}, 25000\}$	$\min\{1800 + 8 \times \text{Capacity}, 5000\}$
Formula for calculating the throughput per disk	$\min\{120 + 0.5 \times \text{Capacity}, 300\}$ MB/s	$\min\{100 + 0.15 \times \text{Capacity}, 140\}$ MB/s
Data durability	99.9999999%	99.9999999%
Single-channel random write access latency	0.5 ms to 2 ms	1 ms to 3 ms
API operation	cloud_ssd	cloud_efficiency
Scenario	Small and medium-sized development and test applications that require high data durability	• Development and test applications • System disks

Purpose-based classification

Cloud disks can be divided by their purposes into system disks and data disks.

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

2.4.2.2.3. Shared Block Storage

Shared Block Storage is 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 Block Storage device can be attached to a maximum of four ECS instances. Shared Block Storage devices can be used only as data disks and must be created separately. Shared access is allowed. You can configure Shared Block Storage devices to be released when the associated ECS instances are released.

Shared Block Storage devices can be divided into the following types based on performance:

- **SSD Shared Block Storage device:** uses SSDs as the storage medium to provide stable and high performance storage that offers enhanced random I/O and data reliability.
- **Ultra Shared Block Storage device:** uses a hybrid SSD and HDD storage medium.

An ECS instance can have up to 16 data disks including cloud disks and Shared Block Storage devices. In other words, the sum of cloud disks and Shared Block Storage devices used as data disks cannot exceed 16.

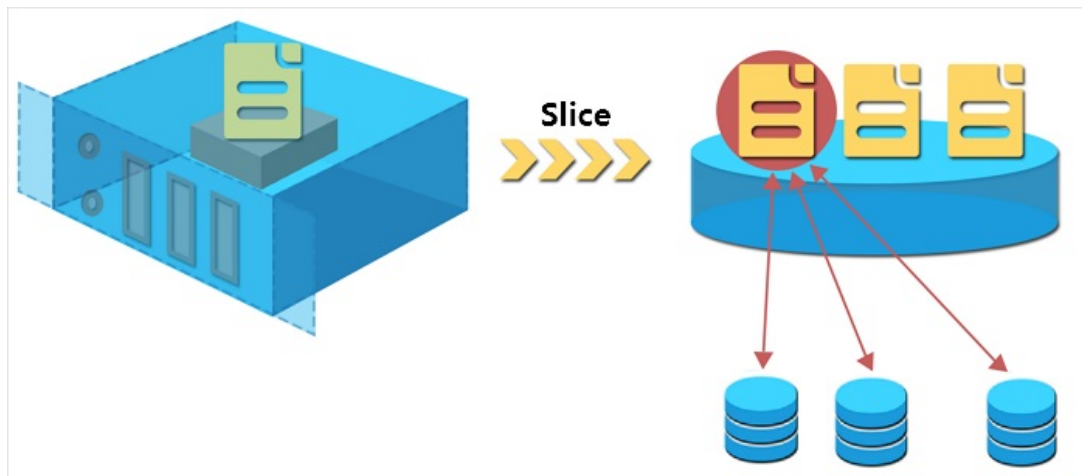
2.4.2.2.4. Triplicate storage

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

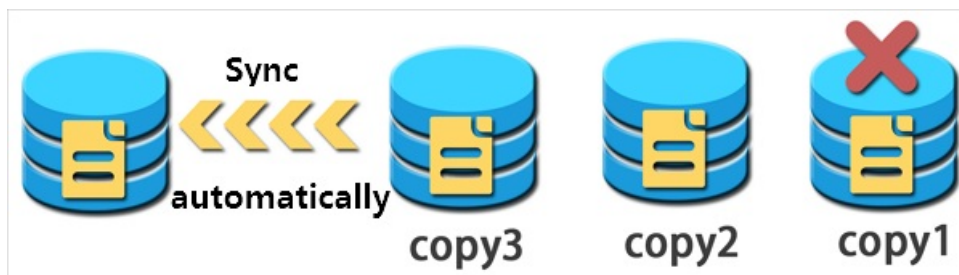
returned.

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.

2.4.2.2.5. Erasure coding

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

What is EC?

EC involves the following concepts:

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

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

Comparison between EC and triplicate storage

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

Item	EC	Triplicate storage
Storage usage	$m/(m+n)$: 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.

2.4.2.3. ECS disk encryption

ECS disk encryption is a simple and secure encryption method that can be used to encrypt new cloud disks.

ECS disk encryption eliminates the need to create or maintain your own key management infrastructure, change existing applications and maintenance procedures, or perform additional encryption operations. Disk encryption does not have any negative impacts on your business processes. The following types of data can be encrypted:

- Data on cloud disks.
- Data transmitted between cloud disks and instances. Data is not encrypted multiple times within the instance operating system.
- All snapshots created from encrypted cloud disks. Such snapshots are encrypted snapshots.

Data transmitted from ECS instances to cloud disks is encrypted on the hosts where the ECS instances are deployed.

All available cloud disks (ultra disks and SSD disks) and Shared Block Storage devices (ultra Shared Block Storage devices and SSD Shared Block Storage devices) in Apsara Stack ECS can be encrypted.

2.4.2.4. Local storage

Local storage, also known as local disks, are disks that reside on the same physical machines as their ECS instances. Local disks provide temporary block storage for instances and are designed for scenarios that require extremely high I/O performance.

Local storage provides block-level data access for instances with high random IOPS, high throughput, and low latency. The reliability of data stored in local disks depends on the reliability of the physical server to which the disks are attached. This is a single point of failure risk which may cause data loss. We recommend that you implement data redundancy at the application layer to ensure the availability of the data.

 **Note** If you storing data on local disks, risks for data persistence may arise, such as when the host server is down. We recommend that you do not use local disks to store data for long periods of time. If no data reliability architecture is available for your applications, we recommend that you use cloud disks or Shared Block Storage devices for your ECS instances.

Local disk types

Apsara Stack provides two types of local disks:

- Local NVMe SSDs: are used together with the gn5 or ga1 instance family.
- Local SATA HDDs: are used together with the d1ne or d1 instance family. This type of local disks is suitable for customers from industries such as Internet and finance that require large storage capacity with storage analysis and offline computing. SATA HDDs satisfy the performance, capacity, and bandwidth requirements of distributed computing models such as Hadoop.

2.4.3. Images

An image is a template for running environments within ECS instances. An image includes an operating system and pre-installed software.

An image works as a copy that stores data from one or more disks. An ECS image may store data from a system disk or from both system and data disks. You can use an image to create an ECS instance or replace the system disk of an ECS instance.

Image types

ECS provides a variety of image types for you to access image resources.

Image description

Type	Description
Public image	Public images provided by Apsara Stack support Windows and many popular versions of Linux operating systems, including: • Windows • CentOS • CoreOS • Debian • Gentoo • FreeBSD • OpenSUSE • SUSE Linux • Ubuntu
Custom image	Custom images created based on your existing physical servers, virtual machines, or cloud hosts. This image type is flexible enough to meet all of your specific business needs.

Obtain an image

You can use one of the following methods to obtain images:

- Create a custom image based on an existing ECS instance.
- Choose an image shared by another Apsara Stack tenant account.
- Import an offline image file to an ECS cluster to generate a custom image.
- Copy a custom image to another region to implement unified deployment of environments and

applications across regions.

Image formats

ECS supports images in the VHD, RAW, and QCOW2 formats. Images in other formats must be converted to the supported formats before they can be run in ECS. For more information, see **Convert image formats** in *ECS User Guide*.

2.4.4. Snapshots

2.4.4.1. 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:

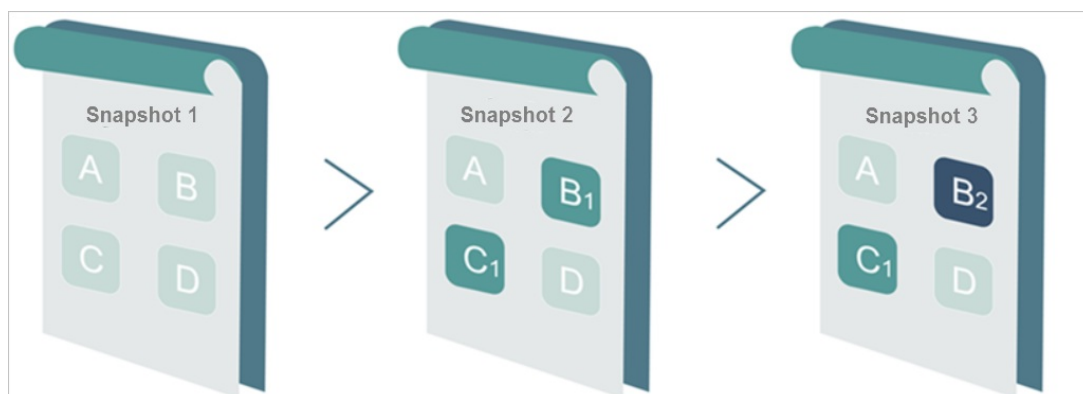
- You may want to use the data of one disk as the basis to write or store data to a different disk. To achieve this, you can create a snapshot for a cloud disk and then create another cloud disk from the snapshot. The new disk contains the basic data of the original disk.
- While cloud disks are a secure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

2.4.4.2. 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.

2.4.4.3. 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.

Item	Traditional snapshot specification	Snapshot 2.0 specification	Benefit	Example
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.

2.4.5. Deployment sets

A deployment set is a tool that allows you to view the physical topology of hosts, racks, and switches and select a deployment policy that best suits the reliability and performance requirements of your business.

There may be increased reliability or performance requirements when you use multiple ECS instances in the same zone.

- **Improve business reliability**

To avoid the impacts caused by the failure of physical hosts, racks, or Switches, multiple copies of application instances must be distributed across different physical hosts, racks, or Switches.

- **Improve network performance**

For scenarios that involve frequent network interactions between instances, lower latency and higher bandwidth can be achieved by aggregating corresponding instances onto a single Switch.

Deployment granularities and policies

- **Deployment granularities**

- Host: indicates physical-server-level scheduling.

- Rack: indicates rack-level scheduling.
- Switch: indicates Switch-level scheduling.

- **Deployment policies**

- LooseAggregation
- StrictAggregation
- LooseDispersion
- StrictDispersion

LooseAggregation and StrictAggregation are intended for higher performance, while LooseDispersion and StrictDispersion are intended for higher reliability.

Granularities and policies lists the deployment policies and business scenarios corresponding to each deployment granularity.

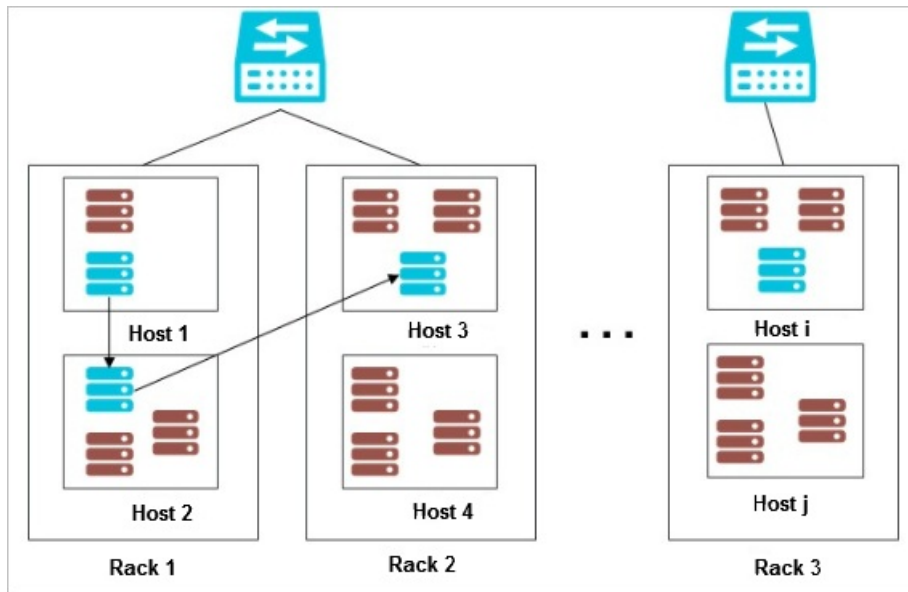
Granularities and policies

Deployment granularity	Deployment policy	Business scenario
Host	StrictDispersion	General purposes
	LooseDispersion	
Rack	StrictDispersion	Big data and databases
	LooseDispersion	Game customers
Switch	StrictDispersion	VPN
	LooseDispersion	Game customers
	StrictAggregation	Big data and databases
	LooseAggregation	Game customers

Typical examples

The following figure shows a typical case where business reliability is improved by using deployment sets. Three ECS instances of a tenant are distributed on three different physical hosts, which are distributed on at least two different racks.

Typical example



Note For more information about the deployment set APIs, see [Deployment sets in ECS Developer Guide](#).

2.4.6. Network and security

2.4.6.1. IP addresses of ECS instances of VPC type

This topic describes the IP address types supported by ECS instances and the corresponding scenarios.

IP address types

ECS instances have the following IP address types:

- **Private IP addresses**

When you create an ECS instance, a private IP address is assigned based on the VPC and the CIDR block of the VSwitch to which the instance belongs.

- **Elastic IP (EIP)**

An EIP is a public IP address. You can apply for an EIP as necessary.

Scenarios

- **Private IP:** A private IP address is used to access the intranet. When creating an instance, you can directly configure the private IP address.

Note If the private IP address is not configured, the system automatically allocates a private IP address for the instance.

- **EIP:** An EIP is used to access the Internet. You can separately bind an EIP to an instance after it has been created. For more information, see [EIP in VPC User Guide](#). EIPs can be applied for and retained long-term. You can bind and unbind an EIP to and from an instance, delete the EIP, or modify its bandwidth.

2.4.6.2. Elastic network interfaces

This topic describes the concepts, scenarios, types, attributes, and limits of elastic network interfaces (ENIs).

What is ENI?

An ENI is a virtual network interface controller (NIC) that can be bound to a VPC-type ECS instance. You can use ENIs to deploy high availability clusters and perform low-cost failovers and fine-grained network management.

Scenarios

ENIs can be used in the following scenarios:

- Deployment of high availability clusters

An ENI can meet the demand for multiple NICs on a single instance within a high availability architecture.

- Low-cost failover

You can unbind an ENI from a failed ECS instance and bind it to another normal instance to quickly redirect traffic destined for the failed instance to the normal instance and immediately restore the service.

- Fine-grained network management

You can configure multiple ENIs for an instance to implement fine-grained network management. For example, you can use some ENIs for internal management and others for Internet business access, so as to isolate management data from business data. You can also configure targeted security group rules for each ENI based on the source IP address, protocols, and ports to implement traffic control.

ENI types

ENIs are classified into two types:

- Primary ENI

A primary ENI is created by default when an instance is created in a VPC. The lifecycle of the primary ENI is the same as that of the instance and you cannot unbind the primary ENI from the instance.

- Secondary ENI

You can create a secondary ENI and bind it to or unbind it from an instance. The maximum number of ENIs that can be bound to a single instance varies with the instance type. For more information, see [Instance families](#).

ENI attributes

The following table describes the attributes of an ENI.

Attribute description

Attribute	Quantity
Primary private IP address	1
MAC address	1

Attribute	Quantity
Security group	1 to 5
Description	1
ENI name	1

Limits

ENIs have the following limits:

- A single account can own up to 100 ENIs within a single region.
- The ECS instance must be in the same zone of the same VPC as the ENI, but does not have to use the same VSwitch.
- For instance types that support ENIs and the number of ENIs supported by each instance type, see [Instance types](#).
- Binding multiple ENIs does not increase the instance bandwidth.

 **Note** The instance bandwidth varies with the instance type.

2.4.6.3. Internal network

If you want to transmit data between two ECS instances within the same region, we recommend that you transmit data over the internal network. ECS instances can also be connected to ApsaraDB RDS, SLB, and OSS over the internal network.

In the internal network, each I/O optimized instance has a shared bandwidth of 10 Gbit/s. The internal network is a shared network and the bandwidth may fluctuate.

 **Note** The actual amount of bandwidth is determined by the physical hardware.

ECS instances can communicate with RDS instances, SLB instances, and OSS buckets within the same region over the internal network.

The following rules apply to VPC-type ECS instances in the internal network:

- Internal communication is permitted by default for instances that belong to the same security group of the same account within the same VPC of the same region. If instances of the same account within the same region belong to different security groups, internal communication can be implemented by authorizing mutual access between the two security groups.
- For instances that belong to the same account and same region but do not belong to the same VPC, you can use Express Connect to implement internal communication.
- The internal IP address of an instance can be modified or changed.
- Virtual IP (VIP) addresses cannot be configured as the internal or public addresses of instances.
- Instances of different network types cannot communicate with each other over the internal network.

2.4.6.4. Security group rules

Security group rules permit or deny Internet or intranet traffic to or from the ECS instances associated with the security group.

You can add or delete security group rules at any time. Changes in security group rules are automatically applied to ECS instances associated with the security group.

Be sure to configure concise security group rules. If you associate an instance with multiple security groups, hundreds of rules may apply to the instance. This may cause connection errors when you access the instance.

2.5. Scenarios

ECS instances can be used either independently as simple web servers or with other Apsara Stack services such as OSS and CDN to provide advanced multimedia solutions. The following sections describe the typical application scenarios of ECS instances:

Official websites for enterprises and simple web applications

Initially, official websites for enterprises do not have high volumes of traffic and only require low-configuration ECS instances to run applications and databases and store files. As your website develops, you can upgrade the configurations and increase the number of ECS instances at any time without the need to worry about insufficient resources during traffic spikes.

Multimedia and high-traffic applications or websites

When you use ECS instances together with OSS, you can store static images, videos, and downloaded packages in OSS to reduce storage costs. You can also use ECS in combination with CDN or SLB to shorten user response time, reduce bandwidth fees, and improve availability.

Applications or websites that have large traffic fluctuations

Some applications and websites may encounter large fluctuations in traffic within a short period of time. ECS provides elastic processing capabilities. The number of ECS instances automatically increases or decreases in response to changes in traffic to meet resource requirements and preserve cost efficiency. ECS can be used in combination with SLB to implement a high availability architecture.

Databases

Databases with high I/O requirements are supported. High-configuration I/O optimized ECS instances can be used together with standard SSDs to support high I/O concurrency and higher data reliability. Alternatively, multiple low-configuration I/O optimized ECS instances can be used in combination with SLB to implement a high availability architecture.

2.6. Terms

This topic describes the basic terms in ECS to help you better understand ECS.

ECS

A simple and efficient cloud computing service that provides elastic processing capabilities and supports operating systems such as Linux and Windows.

instance

An independent resource entity that contains basic resource elements.

security group

A virtual firewall that is used to control the network access of one or more ECS instances and provides stateful inspection and packet filtering. Instances within the same security group are able to communicate with each other, while instances in different security groups are isolated from each other. You can configure the rules of two security groups to authorize mutual access between them.

image

A template for running environments in ECS instances. An image includes an operating system and pre-installed software. Images can be divided into public images and custom images. You can use an image to create an ECS instance or replace the system disk of an ECS instance.

snapshot

Data backup of a disk at a certain point in time. Snapshots consist of automatic snapshots and user-created snapshots.

cloud disk

An independent disk that can be attached to any ECS instance in the same zone of the same region. Cloud disks are divided by performance into ultra disks, SSD disks, and basic disks.

Block Storage

A low-latency and high-reliability persistent random block-level data storage service provided by Apsara Stack for ECS.

throughput

The amount of data successfully transmitted through a network, device, port, virtual circuit, or another facility within a given period of time.

performance test

A world-leading SaaS performance test platform with powerful distributed stress test capabilities. It can simulate real business scenarios with a large number of users to find all application performance problems.

virtual private cloud (VPC)

A virtual private cloud built and customized based on Apsara Stack. Full logical isolation is implemented between VPCs. Users can create and manage cloud service instances, such as ECS instances, SLB instances, and RDS instances in their own VPCs.

internal endpoint

A service connection address for clients that use private IP addresses as their source.

GPU-accelerated instance

A GPU-based computing service used in scenarios such as video decoding, graphics rendering, deep learning, and scientific computation. GPU-accelerated instances provide powerful concurrent and floating point computing capabilities and can process data in real time and at high speed.

3.Container Service for Kubernetes

3.1. What is Container Service?

Container Service provides high-performance, enterprise-class management for scalable Kubernetes-based containerized applications throughout the application lifecycle.

Container Service simplifies the creation and scaling of container management clusters. It integrates Apsara Stack virtualization, storage, network, and security capabilities, providing the optimal environment to run Kubernetes-based containerized applications in the cloud. Alibaba Cloud is a Kubernetes certified service provider, with Container Service being among the first services to pass the Certified Kubernetes Conformance Program. Container Service provides professional container support and services.

3.2. Benefits

Overview

Easy to use

- You can easily create Kubernetes clusters in the Container Service console.
- You can easily upgrade Kubernetes clusters in the Container Service console.

When you use custom Kubernetes clusters, you may need to handle clusters of different versions. Currently, each time you upgrade the clusters, you need to make major adjustments and high operation and maintenance costs are incurred. Container Service allows you to perform rolling upgrades based on images and supports full metadata backups. You can easily roll back clusters to previous versions.

- Allows you to easily scale Kubernetes clusters in the Container Service console.

Kubernetes clusters enable you to quickly scale up or down applications to handle traffic fluctuations in a timely manner.

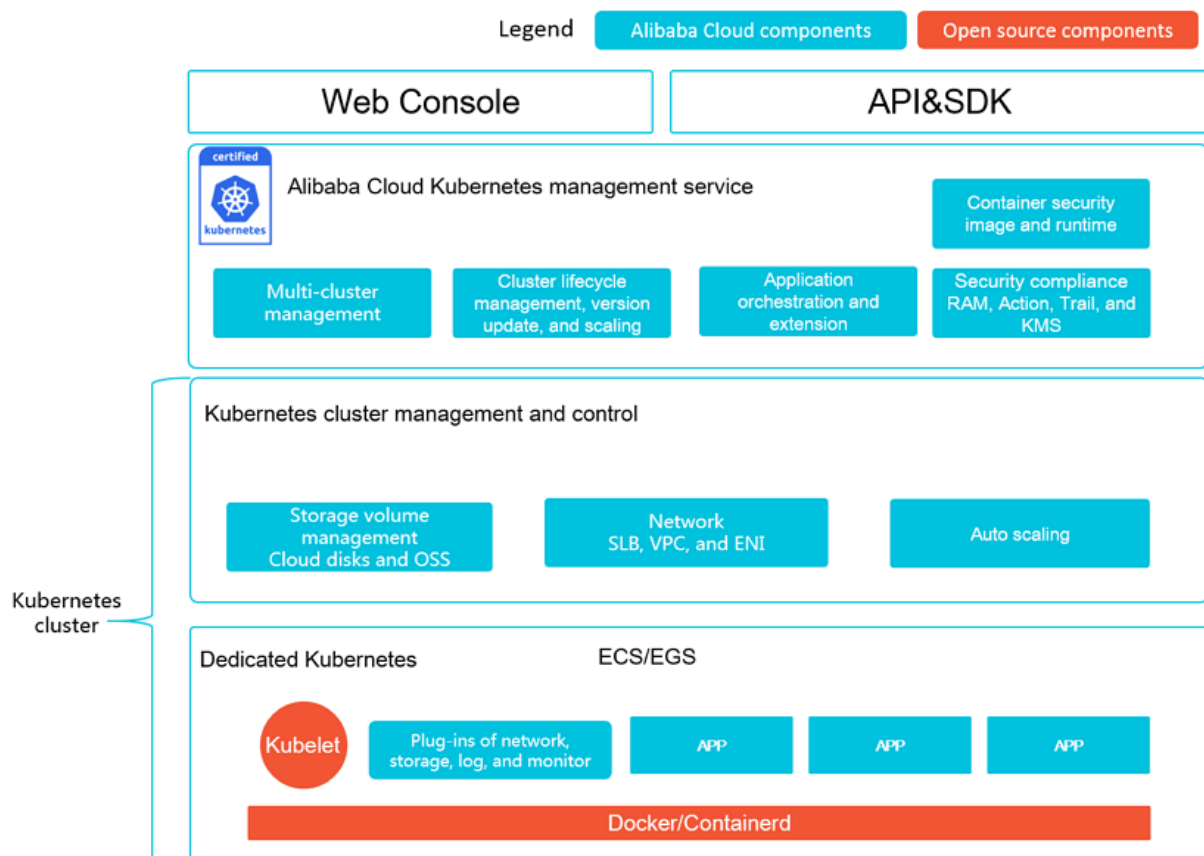
Features

Feature	Description
Network	Supports continuous network integration to optimize network performance.

Feature	Description
Load balancing	Allows you to create public and internal SLB instances. If you use an Ingress to control access to your Kubernetes cluster, frequent service releases may negatively affect the performance of the Ingress and increase the error rate. Container Service allows you to create SLB instances, which provide high availability load balancing and can automatically modify network configurations to suit your business needs. This solution is adopted by a large number of users and has been proven to be a more stable and reliable alternative to Ingresses.
Storage	Supports Apsara Stack cloud disks, Network Attached Storage (NAS), and Block Storage, and provides FlexVolume drivers. Supports seamless integration with cloud storage services for custom Kubernetes clusters that cannot use cloud storage resources.
O&M	• Supports integration with Apsara Stack Log Service. • Supports automatic scaling.
Image repository	• Provides high availability and high concurrency. • Supports accelerated image retrieval. • Supports peer-to-peer image distribution. Custom image repositories may stop responding when millions of clients attempt to pull images at the same time. Container Service provides an image repository system that offers enhanced reliability and reduces O&M and upgrade costs.
Stability	• Dedicated support teams guarantee the stability of containers. • All Linux and Kubernetes versions must pass rigorous testing before they are available to the public. Container Service supports Docker CE and provides a Docker community to help you communicate with other Docker enthusiasts and solve problems. Best practices are provided to help you address issues, such as network interruptions, kernel incompatibilities, or Docker crashes.

Feature	Description
Technical support	- Allows you to quickly upgrade Kubernetes clusters to the latest version. - Provides professional technical support services to help you solve the issues that may occur when you use containers.

3.3. Architecture



Container Service is adapted and enhanced on the basis of native Kubernetes. This service simplifies cluster creation and scaling and integrates Apsara Stack virtualization, storage, network, and security capabilities, providing the optimal environment to run Kubernetes-based containerized applications in the cloud.

Feature	Description
Dedicated Kubernetes mode	Integrated with Apsara Stack virtualization technologies, the service allows you to create dedicated Kubernetes clusters. ECS, Elastic GPU Service (EGS), and ECS Bare Metal instances can all be used as cluster nodes. Instances support a wide range of plug-ins and can be flexible configured to different specifications.

Feature	Description
Alibaba Cloud Kubernetes cluster management and control service	The service provides powerful network, storage, cluster management, scaling, and application extension features.
Alibaba Cloud Kubernetes management service	The service supports secure images and is highly integrated with Apsara Stack Resource Access Management (RAM), Key Management Service (KMS), and logging and monitoring services to provide a secure and compliant Kubernetes solution.
Convenient and efficient use	Container Service for Kubernetes provides services through the Web console, APIs, and SDKs.

3.4. Features

Features

Cluster management

- With the Container Service console, you can easily create a classic dedicated Kubernetes cluster supporting GPU servers within 10 minutes.
- Provides container-optimized OS images as well as Kubernetes and Docker versions that have undergone **stability testing and security enhancement**.
- Supports multi-cluster management, cluster upgrades, and cluster scaling.

Provides end-to-end container lifecycle management

- **Network**

Provides high performance VPC and elastic network interface (ENI) plug-ins optimized for Apsara Stack, boasting 20% increased performance compared with regular network solutions.

Supports container access and throttling policies.

- **Storage**

Container Service is integrated with Apsara Stack disks and OSS, and provides the standard FlexVolume drive.

Supports real-time creation and migration of volumes.

- **Logs**

Provides high-performance log collection integrated with Apsara Stack Log Service.

Supports the integration with third-party open-source logging solutions.

- **Monitoring**

Supports both container-level and VM-level monitoring. Integration with third-party open-source monitoring solutions is supported.

- **Permissions**

Supports cluster-level Resource Access Management (RAM).

Supports application-level permission configuration management.

- **Application management**

Supports phased release and blue-green release.

Supports application monitoring and scaling.

High-availability scheduling policies that allow you to easily handle upstream and downstream delivery processes

- Supports service-level affinity policies and scale-out.
- Provides high availability and disaster recovery across zones.
- Provides cluster and application management APIs to easily implement continuous integration and private system deployment.

3.5. Scenarios

DevOps continuous delivery

Optimized continuous delivery pipeline

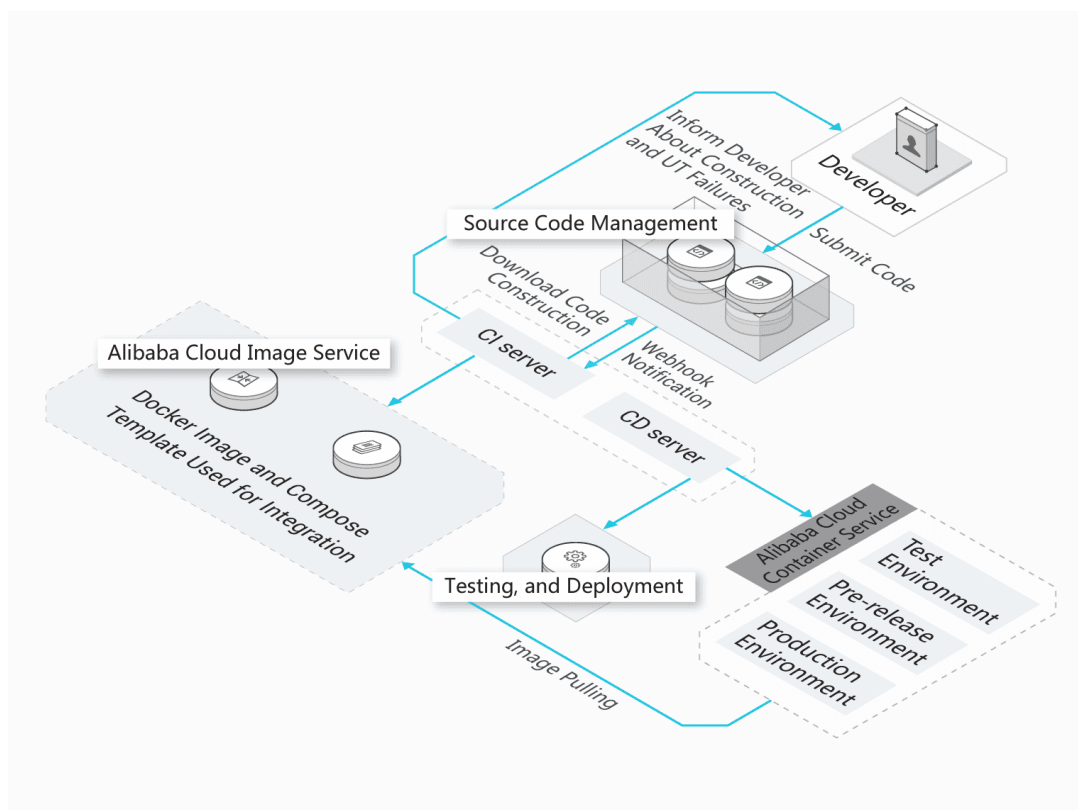
Container Service works with Jenkins to automate the DevOps pipeline, from code submission to application deployments. The service ensures that code is only submitted for deployment after passing automated testing, and provides a better alternative to traditional delivery models that involve complex deployments and slow iterations.

Benefits

- DevOps pipeline automation
Automates the DevOps pipeline, from code updates to code builds, image builds, and application deployments.
- Consistent environment
Allows you to deliver code and runtime environments based on the same architecture.
- Continuous feedback
Provides immediate feedback on each integration or delivery.

Related products and services

ECS + Container Service



Machine learning based on cloud-native technology

Enables rapid application developments with a focus on machine learning

Container Service allows data engineers to easily develop and deploy machine learning applications in heterogeneous computing clusters. Integrated with multiple distributed storage systems, the service supports faster read and write speeds to facilitate the testing, training, and release of data models. You can focus on your core business operations instead of worrying about the deployment and maintenance process.

Benefits

- **Ecosystem support**
Supports mainstream deep learning frameworks, such as TensorFlow, Caffe, MXNet, and Pytorch, and offers optimized features of these frameworks.
- **Quick start and elastic scaling**
Provides machine learning services for development, training, and inference. Supports the startup of training and inference tasks within seconds, and elastic scaling of GPU resources.
- **Easy to use**
Allows you to easily create and manage large-scale GPU clusters and monitor core metrics, such as GPU utilization.
- **Deep integration**
Seamless integration with Apsara Stack storage, logging and monitoring, and security infrastructure capabilities.

Related products and services

ECS/EGS/HPC + Container Service + OSS/NAS/CPFS

Microservices architecture

Agile development and deployment to speed up the evolution of business models

In the production environment, you can split your system into microservices and use Apsara Stack image repositories to store these microservice applications. Apsara Stack can schedule, orchestrate, deploy, and implement phased releases of microservice applications while you focus on feature updates.

Benefits

- Load balancing and service discovery

Forwards layer 4 and layer 7 requests and binds the requests to backend containers.

- Multiple scheduling and disaster recovery policies

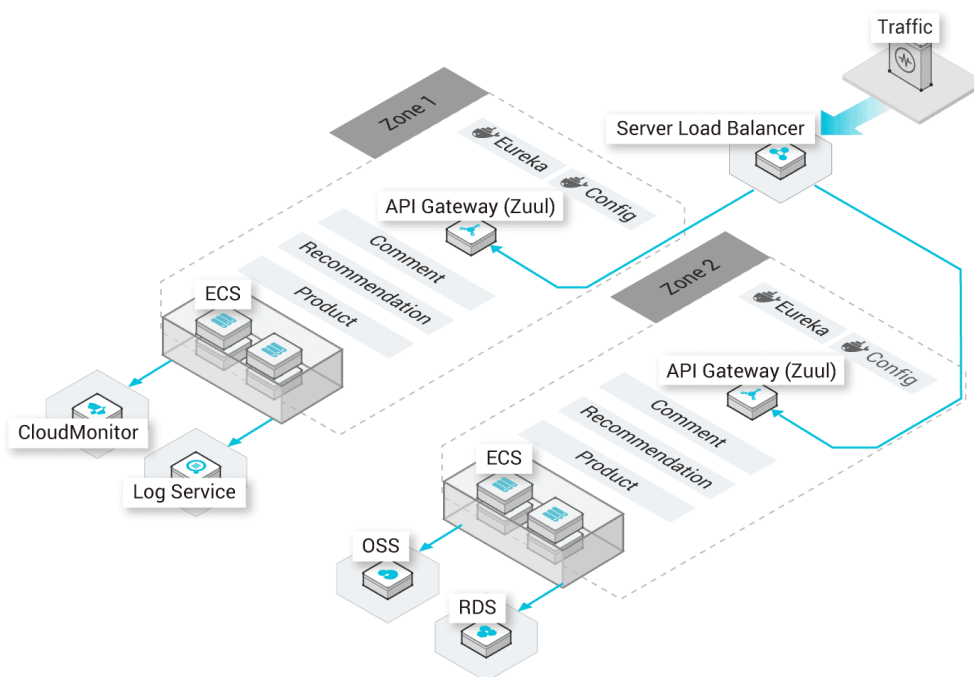
Supports different levels of affinity scheduling policies, and cross-zone high availability and disaster recovery.

- Microservices monitoring and auto scaling

Supports microservice and container monitoring, and microservice auto scaling.

Related products and services

ECS + ApsaraDB RDS + OSS + Container Service



Hybrid cloud architecture

Unified O&M of cloud resources

You can centrally manage cloud and on-premises resources in the Container Service console. Containers hide the differences between infrastructures. This enables you to use the same images and orchestration templates to deploy applications in the cloud and on premises.

Benefits

- Application scaling in the cloud

During peak hours, Container Service can scale up applications in the cloud and forward traffic to the scaled-up resources.

- Disaster recovery in the cloud

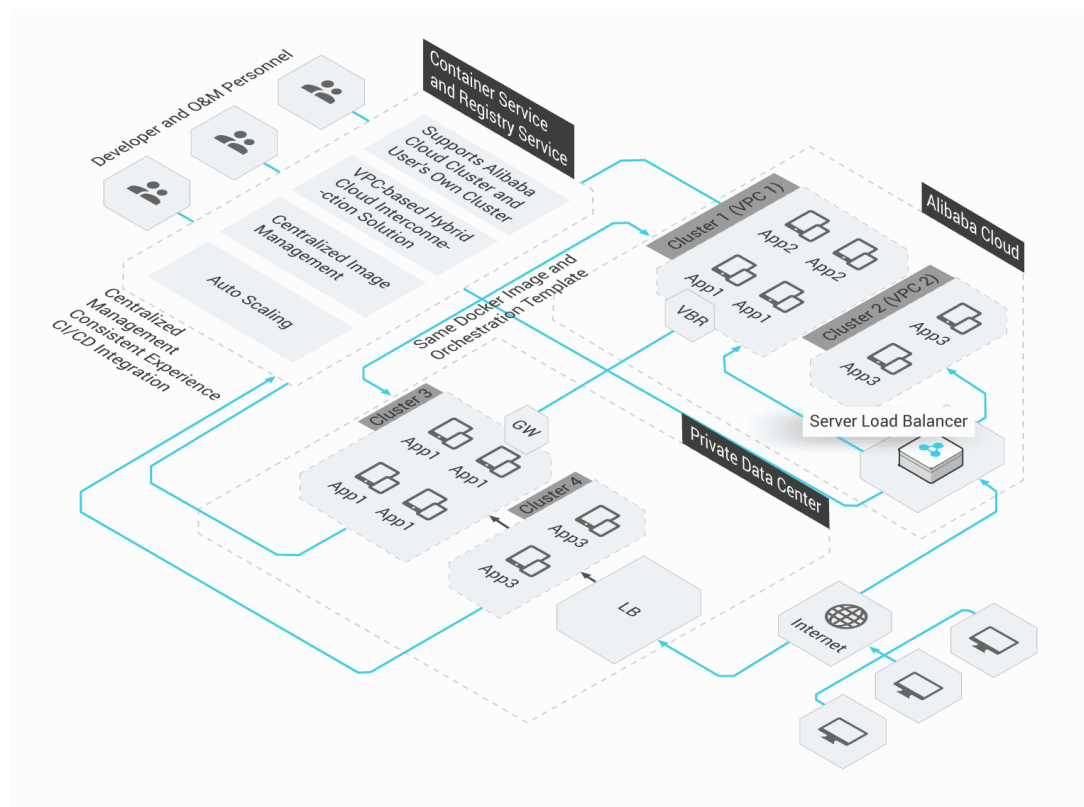
Business systems can be deployed on premises for service provisioning and in the cloud for disaster recovery.

- On-premises development and testing

Applications that are developed and tested on premises can be seamlessly released to the cloud.

Related products and services

ECS + VPC + Express Connect



Automatic scaling architecture

Traffic-based scalability

Container Service enables businesses to auto-scale their resources based on traffic. This prevents traffic spikes from bringing down your system and eliminates idle resources during off-peak hours.

Benefits

- Quick response

Container scale-out can be triggered within seconds when traffic reaches the scale-out threshold.

- Auto scaling

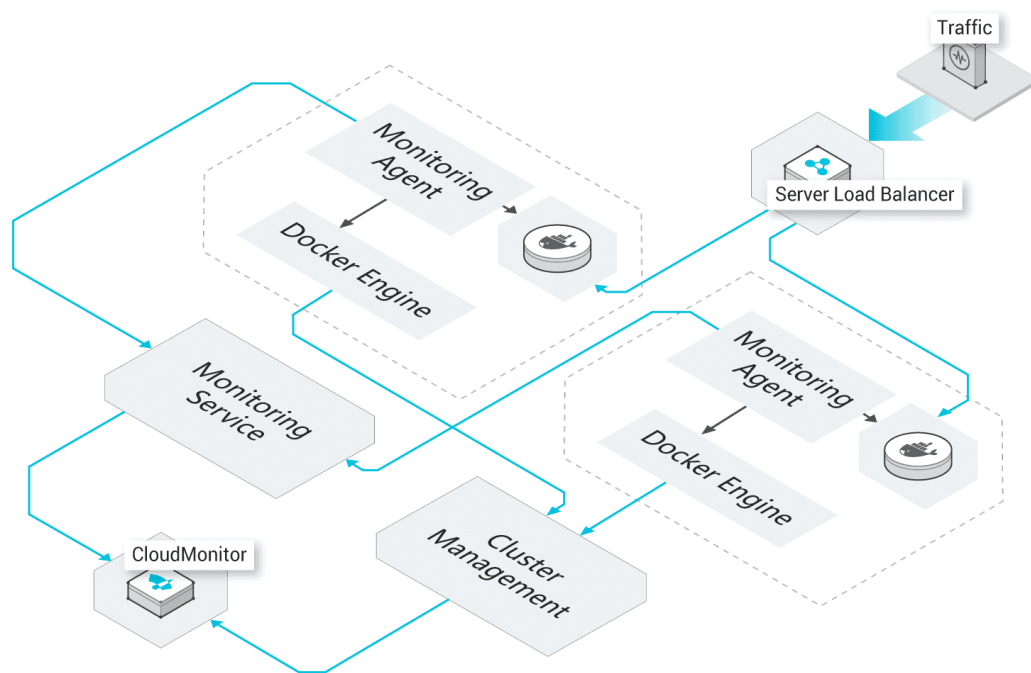
The scaling process is fully automated without human interference.

- Low cost

Containers are automatically scaled in when traffic decreases to avoid resource waste.

Related products and services

ECS + CloudMonitor



3.6. Limits

Limit	Description

Limit	Description
Cluster	- You can create up to 50 clusters across all regions for each Alibaba Cloud account. A cluster can contain up to 100 nodes. To create more clusters or nodes, submit a ticket. - Container Service for Kubernetes (ACK) clusters of Apsara Stack V3.13 are compatible with the ACK clusters of earlier Apsara Stack versions. - After an ACK cluster is created, the following limits apply: - You cannot change the virtual private cloud (VPC). - You cannot change the network plug-in. - You cannot change the storage plug-in. - You cannot migrate applications across namespaces.
Elastic Compute Service (ECS) instance	- ECS instances support both the pay-as-you-go and subscription billing methods. You can change the billing method of ECS instances from pay-as-you-go to subscription in the ECS console. Other resources, such as Server Load Balancer (SLB) instances, support only the pay-as-you-go billing method. - If you want to create subscription ECS instances when you create an ACK cluster, the ECS instances may fail to be created due to resource quota limits and inventory availability. If a subscription ECS instance is created, you cannot release it before the subscription expires. To avoid these issues, we recommend that you create pay-as-you-go ECS instances. You can change the billing method from pay-as-you-go to subscription in the ECS console as required. - Operations such as cluster creation, expansion, and auto scaling are subject to resource quota limits and inventory availability. If the resource quota or inventory is insufficient, you may fail to create nodes. - When you create an ACK cluster, at least two vCPUs and 4 GiB of memory must be specified for ECS instances.

3.7. Terms

cluster

A collection of cloud resources that are required to run containers. Several cloud resources, such as ECS instances, SLB instances, and VPCs, are associated together to form a cluster.

node

A server that has a Docker engine installed and is used to deploy and manage containers. A node can be either an ECS instance or a physical server. The Container Service Agent program is installed on a node and registered to a cluster. The number of nodes in a cluster can be scaled based on your requirements.

container

A runtime instance created from a Docker image. A single node can run multiple containers.

image

A standard packaging format of a containerized application in Docker. An image from the Docker Hub, Alibaba Cloud Container Registry, or your own private registry can be specified to deploy its packaged containerized application. **image ID** An image ID is a unique identifier composed of the image repository URI and image tag. The latest image tag is used for the image ID by default.

Kubernetes terms

node

A worker server in a Kubernetes cluster. A node can be either a virtual server or a physical server. Pods always run on nodes. kubelet runs on each node in a cluster to manage containers in a pod and ensure that they are running properly.

namespace

A method used in Kubernetes to divide cluster resources between multiple users. By default, Kubernetes starts with three initial namespaces: default, kube-system, and kube-public. Administrators can also create new namespaces as required.

pod

The smallest deployable computing unit that can be created and managed in Kubernetes. A pod is a group of one or more containers that share storage and network resources and a common set of specifications for how to run the containers.

Replication Controller (RC)

A feature that monitors running pods to ensure that a specified number of pod replicas are running at any given time. One or more pod replicas can be specified. If the number of pod replicas is smaller than the specified value, an RC starts new pod replicas. If the number of pod replicas exceeds the specified value, the RC stops the redundant pod replicas.

Replica Set (RS)

The upgraded version of RC. Compared with RCs, RSs support more selector types. RS objects are not used independently, but are used as deployment parameters under ideal conditions.

deployment

An update operation performed on a Kubernetes cluster. Deployment is more widely applied than RS. You can use deployments to create, update, or perform rolling updates for services. A new RS is created when you perform a rolling update for a service. A compound operation is carried out to increase the number of replicas in the new RS to the desired value while decreasing the number of replicas in the original RS to zero. This kind of compound operation is better carried out by a deployment than through RS. We recommend that you do not manage or use the RS created by a deployment.

service

The basic operation unit of Kubernetes. It is an abstraction of real application services. Each service has multiple containers that support it. The Kube-Proxy port and service selector determine whether the service request is forwarded to the back-end container, and a single access interface is displayed externally. Back-end operations are invisible to users.

label

A collection of key-value pairs attached to resource objects. Labels are intended to specify identifying attributes of objects that are meaningful and relevant to users, but do not directly imply semantics to the core system. Labels can be attached to objects at creation time, and subsequently added and modified at any time. Each object can have a set of key/value labels, and each key must be unique for a specified object.

volume

Volumes in Kubernetes clusters are similar to Docker volumes. However, they are different in one key aspect. Docker volumes are used to persist data in Docker containers, while Kubernetes volumes share the same lifetime as the pods that enclose them. The volumes declared in each pod are shared by all containers in the pod. The actual back-end storage technology used is irrelevant when you use Persistent Volume Claim (PVC) logical storage. The specific configurations for Persistent Volume (PV) are completed by storage administrators.

PV and PVC

PVs and PVCs allow Kubernetes clusters to provide a logical abstraction over the storage resources, so that the actual configurations of back-end storage can be ignored by the pod configuration logic, and instead completed by the PV configurators. The relationship between PVs and PVCs is similar to that of nodes and pods. PVs and nodes are resource providers which can vary by cluster infrastructure, and are configured by the administrators of a Kubernetes cluster. PVCs and pods are resource consumers that can vary based on service requirements, and are configured by either the users or service administrators of a Kubernetes cluster.

Ingress

A collection of rules that allow inbound access to cluster services. An Ingress can be configured to provide services with externally-reachable URLs, load balance traffic, terminate SSL, and offer name-based virtual hosting. You can request the Ingress by posting Ingress resources to API servers. An Ingress controller is responsible for fulfilling an Ingress, usually with a load balancer. It can also be used to configure your edge router or additional frontends to help handle the traffic.

Related documents

- [Docker glossary](#)
- [Kubernetes concepts](#)

4. Auto Scaling (ESS)

4.1. What is Auto Scaling?

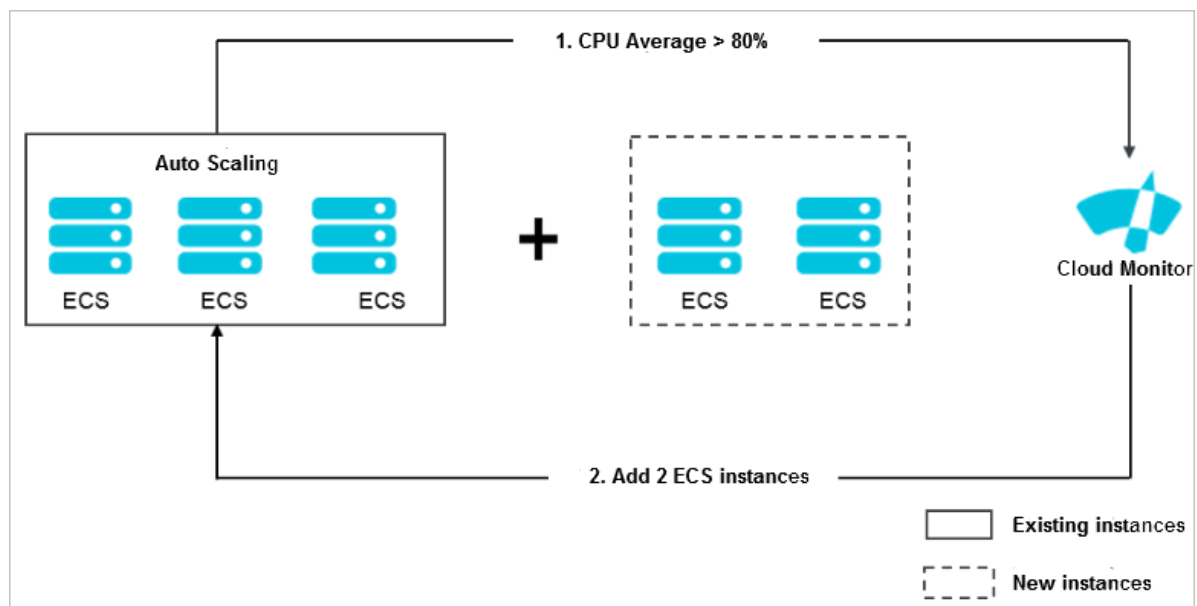
Auto Scaling is a management service that automatically adjusts your elastic computing resources based on your business needs and policies.

When business loads increase, Auto Scaling automatically adds ECS instances based on user-defined scaling rules to ensure sufficient computing capabilities. When business loads decrease, Auto Scaling automatically removes ECS instances to save costs.

Auto Scaling provides the following features:

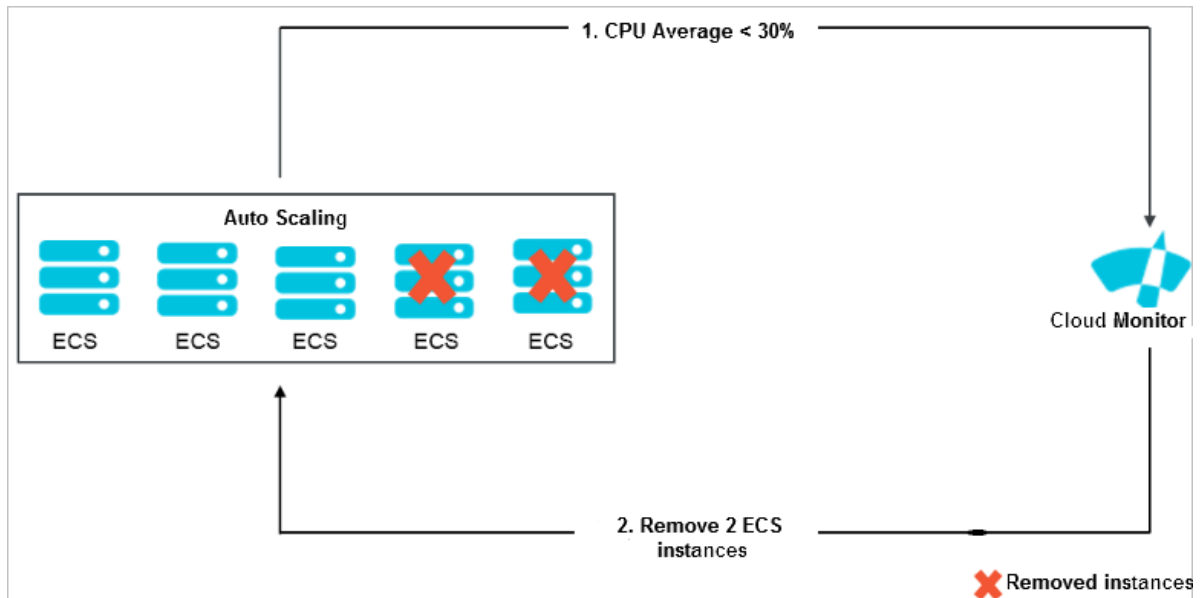
- Scale-out

When business loads surge above normal loads, Auto Scaling automatically increases underlying resources. This helps maintain access speed and ensures that resources are not overloaded. For example, if the CPU utilization of ECS instances exceeds 80%, Auto Scaling scales out ECS resources based on your configured rules. During the scale-out event, Auto Scaling automatically creates and adds ECS instances to a scaling group, and adds the new instances to the backend server groups of the associated SLB instances and the whitelists of the associated ApsaraDB RDS instances. The following figure shows the implementation of a scale-out event.



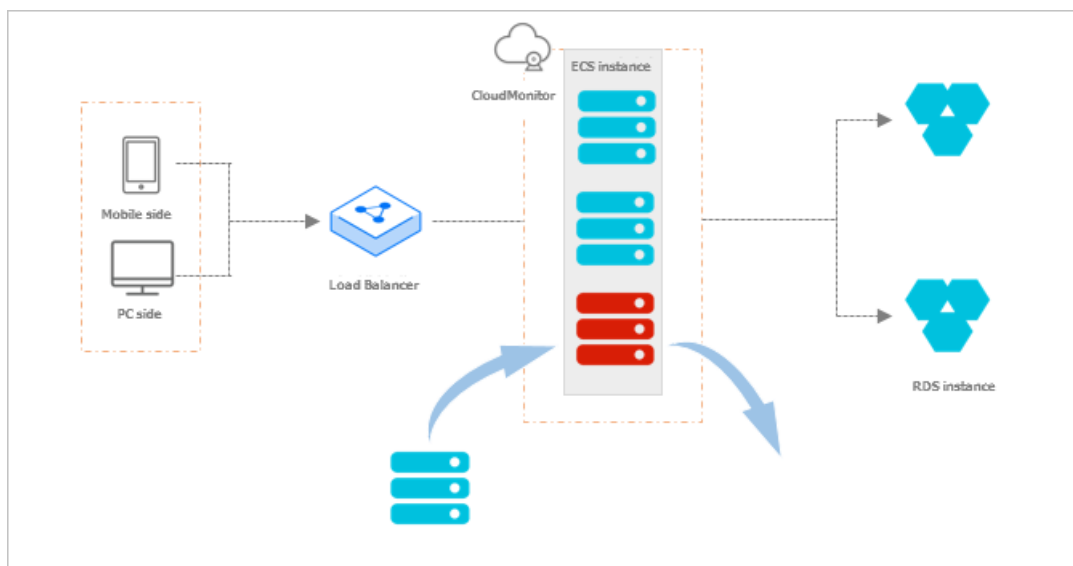
- Scale-in

When your business loads decrease, Auto Scaling automatically releases underlying resources to prevent resource wastage and reduce costs. For example, if the CPU utilization of ECS instances in a scaling group is less than 30%, Auto Scaling automatically scales in ECS instances based on your configured rules. During the scale-in event, Auto Scaling removes ECS instances from the scaling group and also from the backend server groups of the associated SLB instances as well as the whitelists of the associated ApsaraDB RDS instances. The following figure shows the implementation of a scale-in event.



- Elastic recovery

If ECS instances in a scaling group are not in the Running state, Auto Scaling considers the instances to be unhealthy. If an ECS instance is considered unhealthy, Auto Scaling automatically releases the instance and creates a new one. This process is called elastic recovery. It ensures that the number of healthy ECS instances in a scaling group does not fall below the minimum number of ECS instances that you specified for the scaling group. The following figure shows the implementation of elastic recovery.



4.2. Benefits

Compared with manually managing ECS instances, Auto Scaling can help you reduce the infrastructure and O&M costs. This topic describes the benefits of Auto Scaling.

- Automatic scaling of ECS instances

Auto Scaling automatically adds ECS instances during traffic peaks and removes ECS instances when traffic loads drop. This helps reduce infrastructure costs because you pay only for what you actually use.

- Real-time instance monitoring and automatic replacement of unhealthy ECS instances

Auto Scaling performs real-time monitoring on instances and automatically replaces unhealthy instances. This helps save O&M costs.

- Intelligent whitelist management and control, and no user intervention required

Auto Scaling is integrated with Server Load Balancer (SLB) and ApsaraDB RDS (RDS). Auto Scaling automatically manages SLB backend servers and RDS whitelists. This helps save O&M costs.

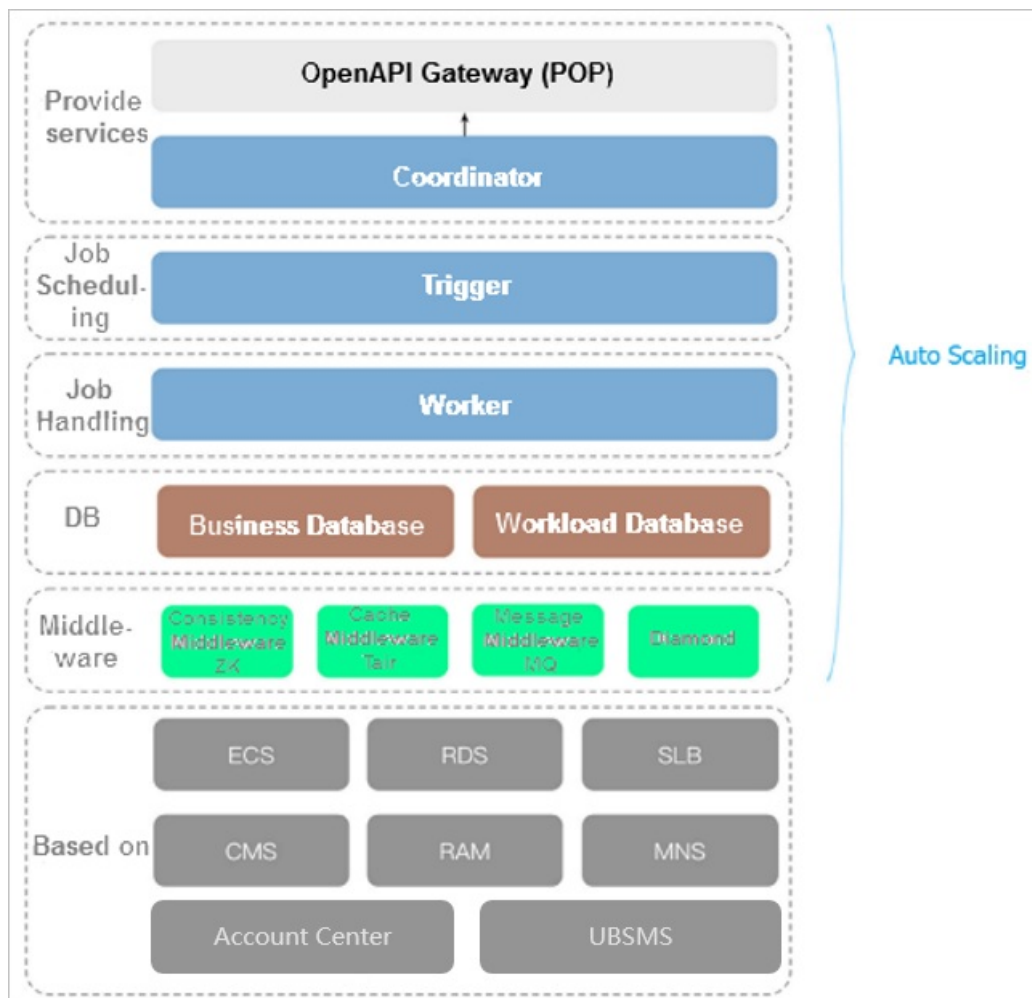
- Various scaling modes for you to mix and match

Auto Scaling allows you to schedule, customize, and fix the minimum number of instances, as well as configure automatic replacement of unhealthy instances. It also provides API operations for you to monitor instances by using external monitoring systems.

4.3. Architecture

This topic describes the architecture of Auto Scaling (ESS) and its components.

The following figure shows the ESS architecture.



The following table describes some of the components in the preceding figure.

Component	Description
Open API Gateway	Provides basic services such as authentication and parameter passthrough.
Coordinator	Serves as the ingress of the ESS architecture. It provides external management and control for services, processes API calls, and triggers tasks.
Trigger	Obtains information from health checks of instances and scaling groups, scheduled tasks, and Cloud Monitor to perform tasks scheduling.
Worker	Functions as the core part of ESS. After ESS receives a task, it processes the entire lifecycle of the task, including splitting the task, executing the task, and returning the execution results.
DB	Includes the business database and workload database.
Middleware layer	ZooKeeper: ensures consistency by implementing distributed locks for Server Controller.
	Tair: provides caching services for Server Controller.
	Message Queue (MQ): provides message queuing services of VM statuses.
	Diamond: manages persistent configurations.

4.4. Features

This topic describes the features of Auto Scaling.

- Adjust the number of ECS instances in various modes

 **Note** Auto Scaling automatically removes ECS instances from scaling groups. Therefore, these instances cannot be used to store application status information such as sessions and related data such as databases and logs. If applications deployed on these ECS instances require data to be stored, you can store the status information on independent ECS instances, store databases in ApsaraDB RDS, and store logs in Log Service.

The scaling modes include:

- Scheduled mode: configures periodic tasks to add or remove ECS instances at a specific point in time, such as 13:00 every day.
- Dynamic mode: dynamically adjusts the number of ECS instances based on the monitoring metrics to ensure that the monitoring metric values are within the expected ranges.

- Custom mode: manually adjusts the number of ECS instances based on your monitoring system statistics.
 - You can manually execute scaling rules.
 - You can manually add or remove ECS instances.
 - After you manually adjust the minimum number of instances (MinSize) and the maximum number of instances (MaxSize), Auto Scaling automatically creates or releases ECS instances to ensure that the number of instances remains between the minimum and the maximum number of instances.
- Fixed-number mode: maintains a fixed number of healthy ECS instances by specifying the MinSize parameter. This mode can be used to ensure daily business availability.
- Health mode: automatically removes or releases ECS instances that are considered unhealthy when they are not in the Running state.
- Multi-mode: combines any of the preceding modes to meet your business requirements. For example, if you predict that business peak hours are between 13:00 and 14:00, you can configure a scaling mode that creates 20 ECS instances at the scheduled time. If you are not sure whether the actual demand during peak hours exceeds the number of scheduled resources, you can configure another scaling mode to handle unexpected business loads. For example, the actual load requires 40 ECS instances.
- Automatically add or remove ECS instances to or from the backend server groups of the associated SLB instances

Auto Scaling automatically adds or removes ECS instances to maintain SLB backend servers and distribute access traffic.
- Automatically add or remove the IP addresses of ECS instances to or from the whitelists of the associated ApsaraDB RDS instances

Auto Scaling automatically adds or removes IP addresses of ECS instances to or from whitelists of associated ApsaraDB RDS instances. This helps maintain the whitelists of associated ApsaraDB RDS instances and control access of ECS instances to the ApsaraDB RDS instances.

4.5. Scenarios

ESS can be used in the following scenarios:

- Video streaming: Traffic loads surge during holidays and festivals. Cloud computing resources must be automatically scaled out to meet the increased demands.
- Live streaming and broadcast: Traffic loads are ever-changing and difficult to predict. Cloud computing resources must be scaled based on CPU utilization, application load, and bandwidth usage.
- Gaming: Traffic loads increase at 12:00 and from 18:00 to 21:00. Cloud computing resources must be scaled out on a regular basis.

4.6. Limits

This topic describes the limits of ESS.

- ESS does not support vertical scaling. It can only scale the number of ECS instances. The CPU, memory, and bandwidth configurations of ECS instances cannot be automatically adjusted.
- The following table describes the quantity limits that are applied to a scaling group.

Item	Quota
Scaling configuration	You can create a maximum of 10 scaling configurations for a scaling group.
Scaling rule	You can create a maximum of 50 scaling rules for a scaling group.
ECS instance	A scaling group can contain a maximum of 1,000 ECS instances.

4.7. Terms

This topic describes the common terms related to Auto Scaling (ESS).

Term	Description
Auto Scaling	Auto Scaling is a management service that automatically adjusts the number of elastic computing resources based on your business demands and policies. It automatically increases ECS instances during high business loads, and automatically releases ECS instances during low business loads.
scaling group	A scaling group is a group of ECS instances that are dynamically scaled based on the configured scenario. You can specify the minimum and maximum numbers of ECS instances in a scaling group, as well as the SLB and Apsara for RDS instances associated with the scaling group.
scaling configuration	Scaling configurations specify the configurations of ECS instances used for automatic scaling.
scaling rule	A scaling rule specifies a specific scaling activity, such as adding or removing N ECS instances.
scaling activity	After a scaling rule is triggered, a scaling activity is executed. A scaling activity shows the changes to the ECS instances in a scaling group.
scaling task	A scaling task is a task that triggers a scaling rule, such as a scheduled task.
cooldown period	The cooldown period indicates a period of time after the completion of a scaling activity in a scaling group. During this period, no other scaling activities can be executed.

5.Resource Orchestration Service (ROS)

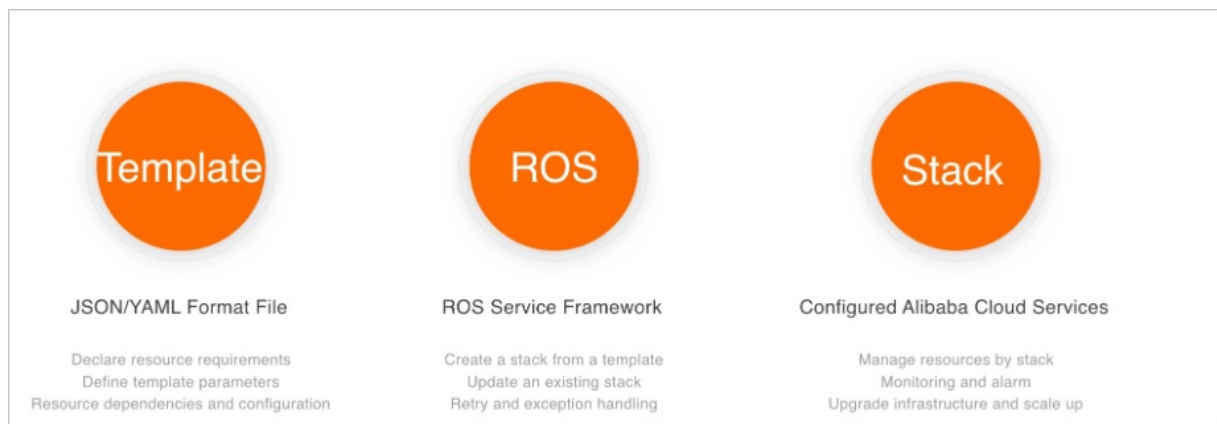
5.1. What is ROS?

Resource Orchestration Service (ROS) is an Apsara Stack service that can simplify the management of cloud computing resources. You can author stack templates based on the template specifications defined in ROS. Within a template, you can define required cloud computing resources such as Elastic Compute Service (ECS) and ApsaraDB RDS instances, and the dependencies between resources. The ROS engine automatically creates and configures all resources in a stack based on a template, which makes automatic deployment and O&M possible.

An ROS template is a readable, easy-to-author text file. You can directly edit a JSON-formatted template or use version control tools such as SVN and Git to control the template and infrastructure versions. You can use APIs and SDKs to integrate the orchestration capabilities of ROS with your own applications to implement Infrastructure as Code (IaC).

ROS templates are also a standardized way to deliver resources and applications. If you are an independent software vendor (ISV), you can use ROS templates to deliver a holistic system or solution that encompasses cloud resources and applications. ISVs can use this method to integrate Apsara Stack resources with their own software systems for centralized delivery.

ROS manages a group of cloud resources as a single unit called a stack. A stack is a group of Apsara Stack resources. You can create, delete, and clone cloud resources by stack.



5.2. Benefits

Resource Orchestration Service (ROS) allows you to model and configure your Apsara Stack resources.

After you create a template that defines your required resources (such as ECS and RDS instances), ROS creates and configures these resources based on the template, facilitating resource management. ROS has the following benefits:

Infrastructure as Code

ROS is an Infrastructure as Code (IaC) solution provided by Alibaba Cloud to quickly implement IaC as a key component of DevOps.

Fully managed automation service

ROS is a fully managed service and does not require you to purchase the resources that are used to maintain your templates, allowing you to focus on maintaining the resources of your business and the template specifications. When you need to create multiple projects that are distributed across multiple stacks, managed automation of the creation process enables you to complete tasks faster. We recommend that you use ROS API operations to maintain stacks and use source code versioning software such as Git and SVN for centralized management of templates.

Repeatable deployment

You can use the same templates to deploy resources to the development, test, and production environments. You can set parameters to different values for different environments. For example, you can set the number of ECS instances in the test environment to 2 and the number of ECS instances in the production environment to 20. You can also use the same templates to deploy resources to multiple regions. This improves the efficiency of multi-region deployment.

Standardized deployment

In practice, subtle differences in different environments often lead to complicated management and high costs, prolong troubleshooting time, and interfere with the normal operation of your business. By using ROS for repeated deployment, you can standardize deployment environments, minimize the differences between different environments, and build environment configurations into templates. A rigorous management process similar to code implementation can ensure standardized deployment practices.

Unified authentication, security, and audit

Compared with other similar services, ROS provides better integration with other Apsara Stack services. Integration with Resource Access Management (RAM) provides unified authentication, eliminating the need to establish a separate user authentication system. Operations on all cloud services are called through APIs. You can use ActionTrail to review all O&M operations, including operations on ROS.

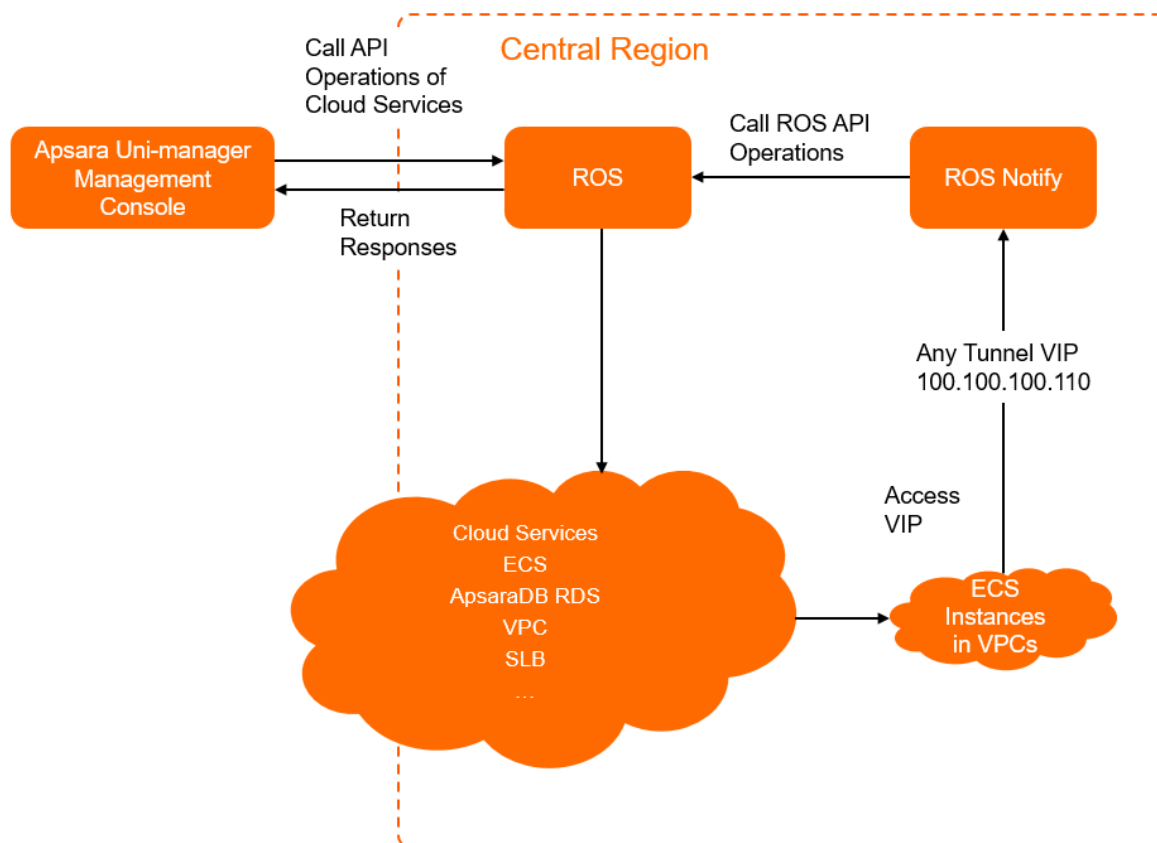
5.3. Architecture

This topic introduces the architecture of Resource Orchestration Service (ROS). You can use ROS by means of the Elastic Compute Service (ECS) console, API operations, and SDKs.

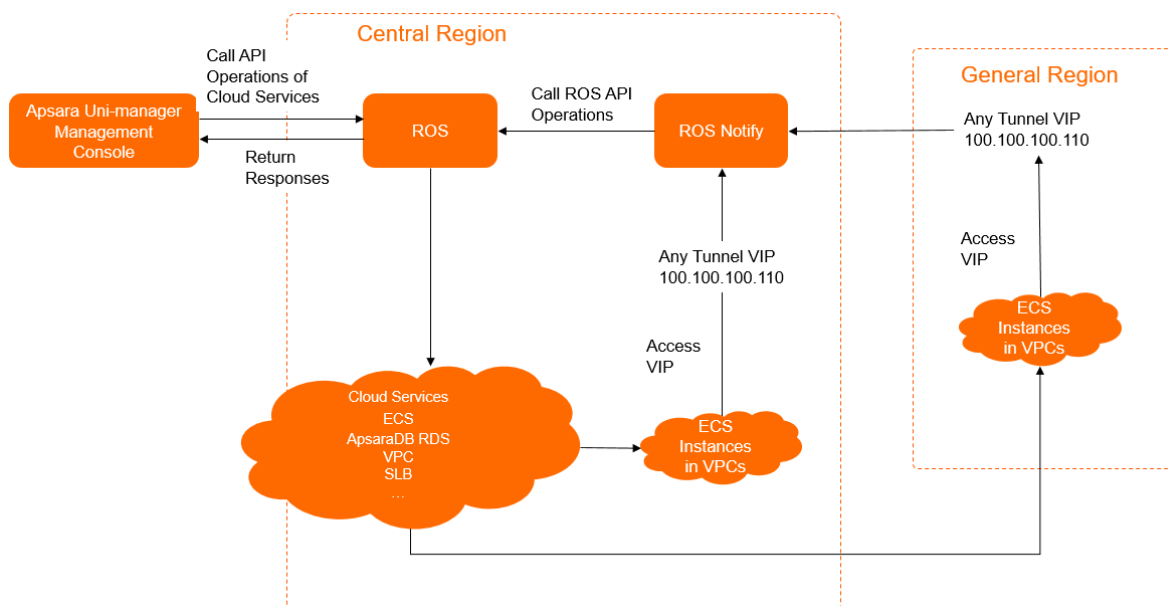
ROS supports the following cloud services: ECS, ApsaraDB RDS, ApsaraDB for MongoDB, ApsaraDB for Redis, ApsaraDB for Memcache, Server Load Balancer (SLB), Object Storage Service (OSS), Virtual Private Cloud (VPC), Elastic IP Address (EIP), Auto Scaling (ESS), Log Service, and Resource Access Management (RAM).

ROS supports single-region and multi-region deployment. The following figure shows the ROS architecture.

- Single-region deployment



- Multi-region deployment



5.4. Features

This topic describes the features of Resource Orchestration Service (ROS). The ROS engine automatically creates and configures all resources in a stack based on a template, which makes automatic deployment and Operations and Maintenance (O&M) possible.

ROS is an important service in cloud computing. You can define your cloud infrastructure as ROS templates and deploy them to the cloud from anywhere at any time to implement Infrastructure as Code (IaC). Compared with API calls of services, ROS allows you to process business resources in a more efficient way. Resources are created in stacks in ROS. You can also access these resources in separate consoles. You can manage stacks and their resources in the ROS console.

Create a stack

ROS manages a group of cloud resources as a single unit called a stack. A stack is a group of Alibaba Cloud resources. You can create, delete, and update cloud resources by using stacks. In DevOps scenarios, you can clone the development, test, and production environments. This simplifies the overall migration and scaling of applications.

Update a stack

If you only need to modify the current template and configurations of a specified stack but do not need to change the region where the stack resides, update the stack. You can modify the template and parameter configurations of the stack by updating the stack.

Recreate a stack

If you need to modify the current template and configurations of a specified stack and change the region where the stack resides, recreate the stack.

Delete a stack

You can delete a stack that is no longer needed. You can choose whether to retain or release resources when you delete a stack.

5.5. Scenarios

Resource Orchestration Service (ROS) is applicable to a wide range of scenarios. It helps migrate your business to the cloud, and supports on-demand batch deployment and business environment distribution. ROS uses approved templates to deploy cloud environments, which helps meet IT compliance requirements and minimize financial risks.

Migration of business to the cloud

The best practices provided by Alibaba Cloud are employed to deploy all resources defined in a solution template and optimize the cloud architecture. No professional IT skills or experience in cloud architecture design are required.

On-demand batch deployment

Templates are used to deploy multiple application runtime environments in business expansion or DevOps scenarios.

Business environment distribution

In centralized IT management scenarios, standardized environment distribution is conducted across regions and accounts to meet the business needs of each organization and team.

Cloud environment management

Approved templates are used to deploy cloud environments, which helps meet IT compliance requirements and minimize financial risks.

5.6. Limits

This topic describes the limits of Resource Orchestration Service (ROS).

When you use ROS, take note of the following items:

- Each stack can contain up to 200 resources.
- Each user can create up to 50 stacks.
- Each template file can be up to 512 KB in size.

5.7. Terms

This topic introduces the terms related to Resource Orchestration Service (ROS).

Term	Description
template	A JSON or YAML file encoded in UTF-8. You can save template files with extensions such as .json, .yaml, .template or .txt. Templates are used to create stacks and serve as the blueprint for the underlying infrastructure and architecture. Templates define the configurations and dependencies of Alibaba Cloud resources.
stack	A collection of ROS resources that you can manage as a single unit. You can create, update, or delete a stack to create, update, or delete a group of resources.

6.Object Storage Service (OSS)

6.1. What is OSS?

Object Storage Service (OSS) is a secure, cost-effective, and highly reliable cloud storage service provided by Alibaba Cloud. It enables you to store a large amount of data in the cloud.

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 the network at any time.

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.

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 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.

6.2. Benefits

OSS provides secure, cost-effective, and high-durability services for you to store large amounts of data in the cloud. This topic compares OSS with the traditional user-created server storage to show the benefits of OSS.

Advantages of OSS over user-created server storage

Item	OSS	User-created server storage
Reliability	Automatically stores multiple copies of data for backup.	• Prone to errors due to low hardware reliability. If a disk has a bad sector, data may be lost. • Manual data restoration is complex and requires a lot of time and technical resources.

Item	OSS	User-created server storage
Security	- Provides multi-level security protection for enterprises. - Provides resource isolation mechanisms for multiple tenants and supports geo-disaster recovery. - Provides various authentication and authorization mechanisms. It also provides features such as whitelists, hotlink protection, RAM, and Security Token Service (STS) for temporary access.	- Requires additional scrubbing devices and black hole policy-related services. - Requires a separate security mechanism.
Data processing	Provides Image Processing (IMG).	Data processing capabilities must be purchased and separately deployed.

More benefits of OSS

- Ease of use

OSS provides standard RESTful API operations, a wide range of SDKs, client tools, and the OSS console. You can upload, download, retrieve, and manage large amounts of data used for websites and mobile apps in the same way you use regular file systems.

- Streaming writes and reads are supported, which can be used in business scenarios where you must simultaneously read and write videos and other large objects.
- Lifecycle management is supported. You can configure lifecycle rules to delete expired objects in batches.

- Powerful and flexible security mechanisms

Flexible authentication and authorization mechanisms are available. OSS provides STS and URL-based authentication and authorization mechanisms, whitelists, hotlink protection, and RAM.

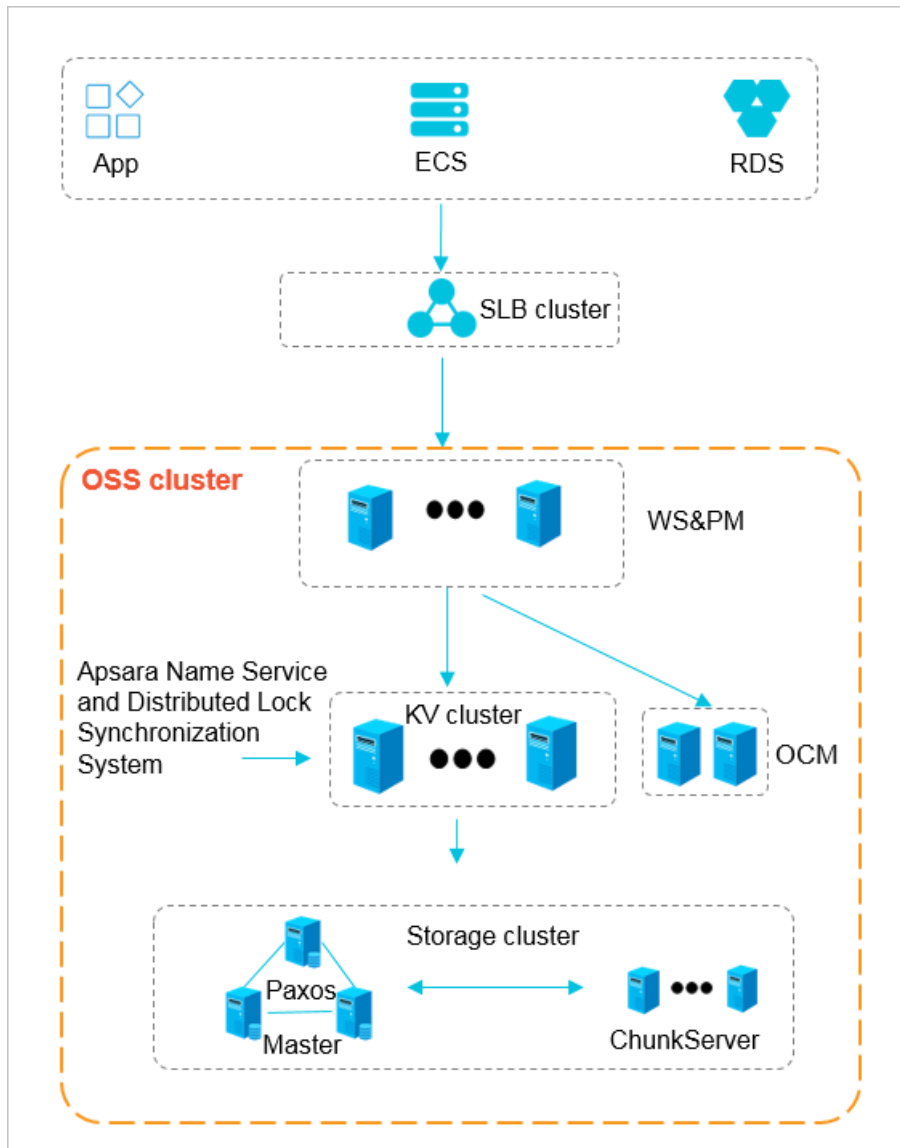
- Rich image processing features

OSS supports the conversion between objects in multiple formats such as JPG, PNG, BMP, GIF, WebP, and TIFF. OSS also supports various operations for image objects, such as thumbnails, cropping, watermarking, and resizing.

6.3. Architecture

OSS is a storage solution that is built on the Apsara system. It is based on the infrastructure such as Apsara Distributed File System and SchedulerX. The infrastructure provides OSS and other Alibaba Cloud services with importance features such as distributed scheduling, high-speed networks, and distributed storage. The following figure shows the OSS architecture.

OSS architecture



- WS & PM: the protocol layer that receives and authenticates the request sent by using a RESTful protocol. If the authentication is successful, the request is forwarded to KVEngine for further processing. If the authentication fails, an error message is returned.
- KV cluster: used to process structured data, including reading and writing data based on object names. The KV cluster also supports sporadic bursts of requests. When a service has to run on a different physical server due to a change to the service coordination cluster, the KV cluster can coordinate and find the access point.
- Storage cluster: Metadata is stored in the master node. A distributed message consistency protocol of Paxos is adopted between Master nodes to ensure the consistency of metadata. This method ensures efficient distributed storage of and access to objects.

6.4. Terms

This topic describes several basic terms used in OSS.

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 globally unique within OSS. Bucket names cannot be changed after the buckets are created.
- A bucket can contain an unlimited number of objects.

Strong consistency

A feature requires that object operations in OSS be atomic, which indicates that operations can only either succeed or fail. There are no 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 have been 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 you delete an object, the object and its copies no longer exist.

Similar to traditional storage devices, modifications are immediately visible in OSS while 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, 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 API operation must be called to append an object, and the generated object is different from objects uploaded by using other methods. To modify even a single byte, 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.

6.5. Features

This topic lists the common features of OSS.

Before you start to use OSS, we recommend that you have a good understanding of basic terms used in OSS, such as bucket, object, region, and endpoint. For more information, see [Terms](#).

The following table describes features of OSS.

OSS features

Parameter	Feature	Description
Bucket	Create buckets	Before you upload an object to OSS, you must create a bucket to store the object.
	Delete buckets	If you no longer use a bucket, delete it to avoid further fees.
	Modify bucket ACL	OSS supports ACL for access control. You can configure the ACL of a bucket when you create it or modify the ACL of a created bucket.
	Configure static website hosting	You can configure static website hosting for your bucket and access this static website through the bucket domain name.
	Configure hotlink protection	To prevent additional fees caused by unauthorized access to the data in your bucket, you can hotlink protection for your buckets based on the Referer field in HTTP requests.
	Manage CORS	OSS provides cross-origin resource sharing (CORS) over HTML5 to implement cross-origin access.

Parameter	Feature	Description
	Configure lifecycle rules	You can define and manage lifecycle rules for all or a subset of objects in a bucket. You can configure lifecycle rules to manage multiple objects and automatically delete parts.
	Configure server-side encryption	OSS can perform server-side encryption on data uploaded to buckets to secure data.
Object	Upload objects	You can upload any type of objects to a bucket.
	Create folders	You can manage OSS folders the way you manage folders in Windows.
	Search for objects	You can search for objects whose names contain the same prefix in a bucket or folder.
	Obtain object URLs	You can obtain the URL of an object to share or download the object.
	Delete objects	You can delete a single object or multiple objects.
	Delete folders	You can delete a single folder or multiple folders.
	Modify object ACL	You can configure ACL when you upload an object and modify the ACL after you upload the object.
	Manage parts	You can delete all or some parts from a bucket.
API	API	OSS supports RESTful API operations and provides examples.
SDK	SDK	OSS supports development based on SDKs for various programming languages and provides examples.

6.6. Scenarios

This topic describes the application scenarios of OSS.

Massive storage for image, audio, and video applications

OSS can be used to store large amounts of data, such as images, audio and video data, and logs. OSS supports various devices. Websites and mobile applications can directly read or write OSS data. OSS supports file writing and streaming writing.

Dynamic and static content separation for websites and mobile applications

By using the BGP bandwidth, you can download data from OSS with an ultra-low latency.

Offline data storage

OSS provides storage with low cost and high availability. Therefore, you can use OSS to store enterprise data that needs to be archived offline for a long period.

6.7. Limits

This topic describes the limits and performance metrics of OSS.

Item	Limit
Bucket	- You can create a maximum of 100 buckets. - After a bucket is created, its name and region cannot be modified.
Object upload	- 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. - OSS traffic is forwarded through SLB and has the following limits: - By default, a virtual IP address (VIP) is configured for SLB and the maximum throughput for OSS is 1.25 GB/s. - The maximum throughput for each OSS node is 300 MB/s. In scenarios where only stable and frequent write operations are continuously performed, the maximum throughput for each OSS node is 100 MB/s.
Object deletion	- 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.

7. Tablestore

7.1. What is Tablestore?

Tablestore is a NoSQL database service independently developed by Alibaba Cloud. Tablestore is a proprietary software program that is certified by the relevant authorities in China. Tablestore is built on the Apsara system of Alibaba Cloud, and can store large amounts of structured data and allow real-time access to the data.

Tablestore provides the following features:

- Offers schema-free data storage. You do not need to define attribute columns before you use them, or perform table-level changes to add or delete attribute columns. You can set the time to live (TTL) parameter for a table to manage the lifecycle of data. Expired data is deleted from the table.
- Uses the multi-node cluster architecture. Each management node on the platform implements a high availability mechanism. Therefore, the failure of a service node within a cluster does not affect the overall operating of businesses.
- Adopts the triplicate technology to keep three copies of data across different servers in different racks. A cluster can support single storage type instances (SSD only) or mixed storage type instances (SSD and SATA) to meet different budget and performance requirements.
- Adopts a fully redundant architecture to prevent single points of failure (SPOFs). The support for smooth online upgrades, hot cluster upgrades, and automatic data migration enables you to dynamically add or remove nodes for maintenance without incurring service interruptions. The concurrent read/write throughput and storage capacity can be linearly scaled. Each cluster can have no less than 500 hosts.
- Supports highly concurrent read/write operations. Concurrent read/write capabilities can be scaled out as the number of hosts increases. The read/write performance is indirectly related to the amount of data in a single table.
- Supports identity authentication and multi-tenancy. Comprehensive access control and isolation mechanisms are provided to safeguard your data. VPC and access over HTTPS are supported. Provides multiple authentication and authorization mechanisms so that you can define access permissions on individual tables and operations.

7.2. Benefits

Tablestore provides the following benefits:

Scalability

- Tablestore imposes no upper limit on the amount of data that can be stored in tables. When the amount of data increases, Tablestore adjusts partitions to provide more storage space for tables and improve the capability of handling access request bursts.
- Tablestore supports CPUs, disks, memory, and network interface controllers (NICs) of different specifications in a single-component cluster without affecting cluster running performance. This ensures maximum compatibility with existing devices.

High performance

If you use a high-performance instance, its average access latency of single rows is measured in single-digit milliseconds. The read/write performance is not affected by the size of data in a table.

Data reliability

- Tablestore provides high data reliability. It stores multiple data copies and restores data when any of the copies become invalid.
- Tablestore supports automatic fault tolerance for the failures of server hard disks in a cluster and supports hot swapping of hard disks. In the event of a hard disk failure, services can be restored within two minutes.
- Tablestore supports full or incremental backup and data recovery from storage.
- Tablestore supports the backup between data clusters in different data centers. The backup process is visualized.
- Tablestore supports the backup and restoration of the metadata, files, and tables of key components.

High availability

Tablestore uses automatic failure detection and data migration to shield applications from host- and network-related hardware faults, which provides high availability for your applications.

Ease of management

- Tablestore automatically performs complex O&M tasks, such as the management of data partitions, software and hardware upgrades, configuration updates, and cluster scale-out.
- You can store audit logs to Log Service and download logs from Log Service. This facilitates the long-term storage and management of audit logs.

Access security

- Tablestore provides multiple permission management mechanisms. It verifies and authenticates the identity of each application request to prevent unauthorized data access, which improves data security.
- Tablestore supports the management of data access permissions, including logon permissions, table creation permissions, read and write permissions, and whitelist-related permissions.
- Tablestore allows you to use the Apsara Uni-manager Management Console to manage administrative permissions, including administrator classification. You can use the Apsara Uni-manager Management Console to manage user permissions in a centralized manner. You can manage the access control features of all components in the system. You can also block common users from querying access control details and simplify access control for administrators. This improves the usability of access control.

Strong consistency

Tablestore ensures high data consistency for data writes. After a write operation succeeds, three replicas are written to a disk. Applications can read the latest data immediately.

Flexible data models

Tablestore tables do not require a fixed format. Each row can contain a different number of columns. Tablestore supports multiple data types, including Integer, Boolean, Double, String, and Binary.

Monitoring integration

You can log on to the Tablestore console to obtain monitoring information in real time, including the requests per second and average response latency.

Multitenancy

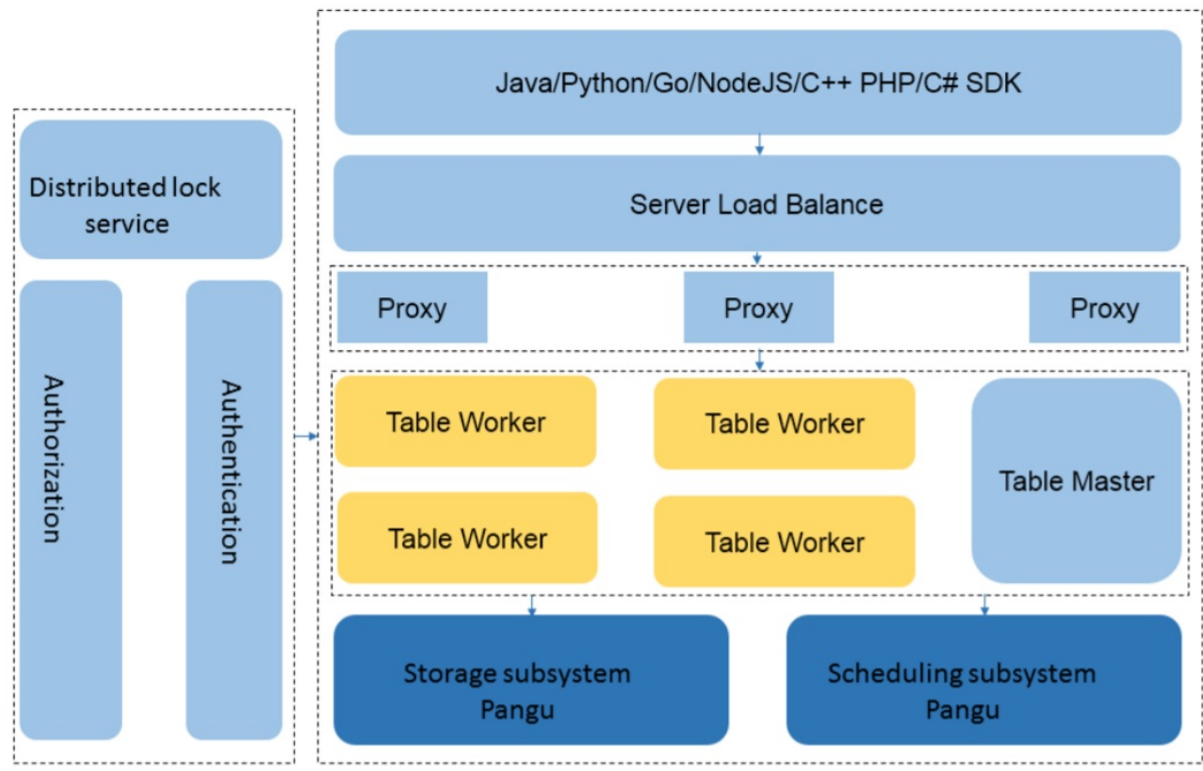
- Isolation: allows tasks of multiple tenants (projects) to be submitted to different queues and run separately. Resources are isolated among tenants.
- Permission: allows you to manage tenants in a centralized manner, dynamically configure and manage tenant resources, isolate resources, view statistics for resource usage, and manage tenants at multiple levels in the console.
- Scheduling: supports multi-tenant scheduling of multiple clusters and multiple resource pools.

7.3. Architecture

This topic describes the Tablestore architecture.

The architecture of Tablestore is referenced from Bigtable (one of the three core technologies of Google) and uses the log-structured merge-tree (LSM) storage engine to provide high write performance. The performance of primary key-based single-row queries and range queries is stable and predictable. The performance is not affected by the volume of data and access concurrency.

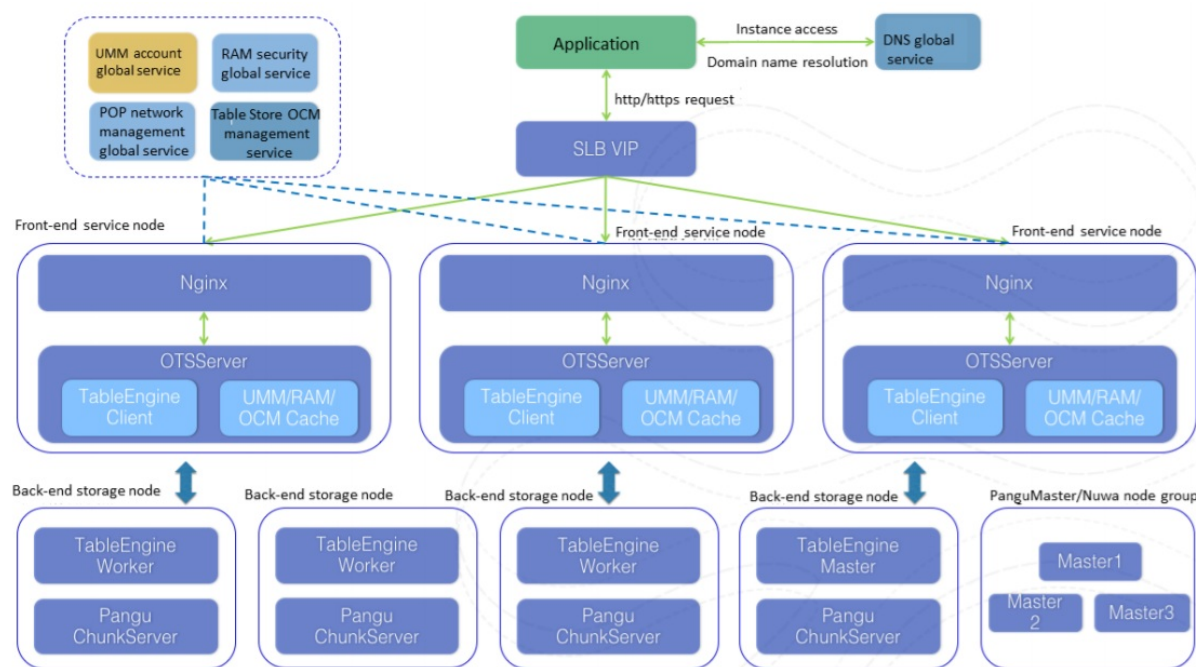
The following figure shows the basic architecture of Tablestore.



- The top layer is the protocol access layer. Server Load Balancer (SLB) distributes user requests to various proxy nodes. The proxy nodes receive requests that are sent by using the RESTful protocol and implement security authentication.
 - If the authentication succeeds, the user requests are forwarded to the corresponding data engine based on the value of the first primary key column for further operations.
 - If the authentication fails, error information is returned to the user.
- Table Worker is the data engine layer that processes structured data. It uses a primary key to search for or store data. Table Worker supports large-scale access request bursts.

- The bottom layer is the persistent storage layer. Apsara Distributed File System is deployed at this layer. Metadata is stored on masters. A distributed message consistency protocol (or Paxos) is adopted between masters to ensure the metadata consistency. This way, efficient distributed file storage and access are achieved. This method ensures that three copies of data are stored in the system and that the system can recover from any hardware or software fault.

The following figure shows the detailed architecture of Tablestore.



7.4. Scenarios

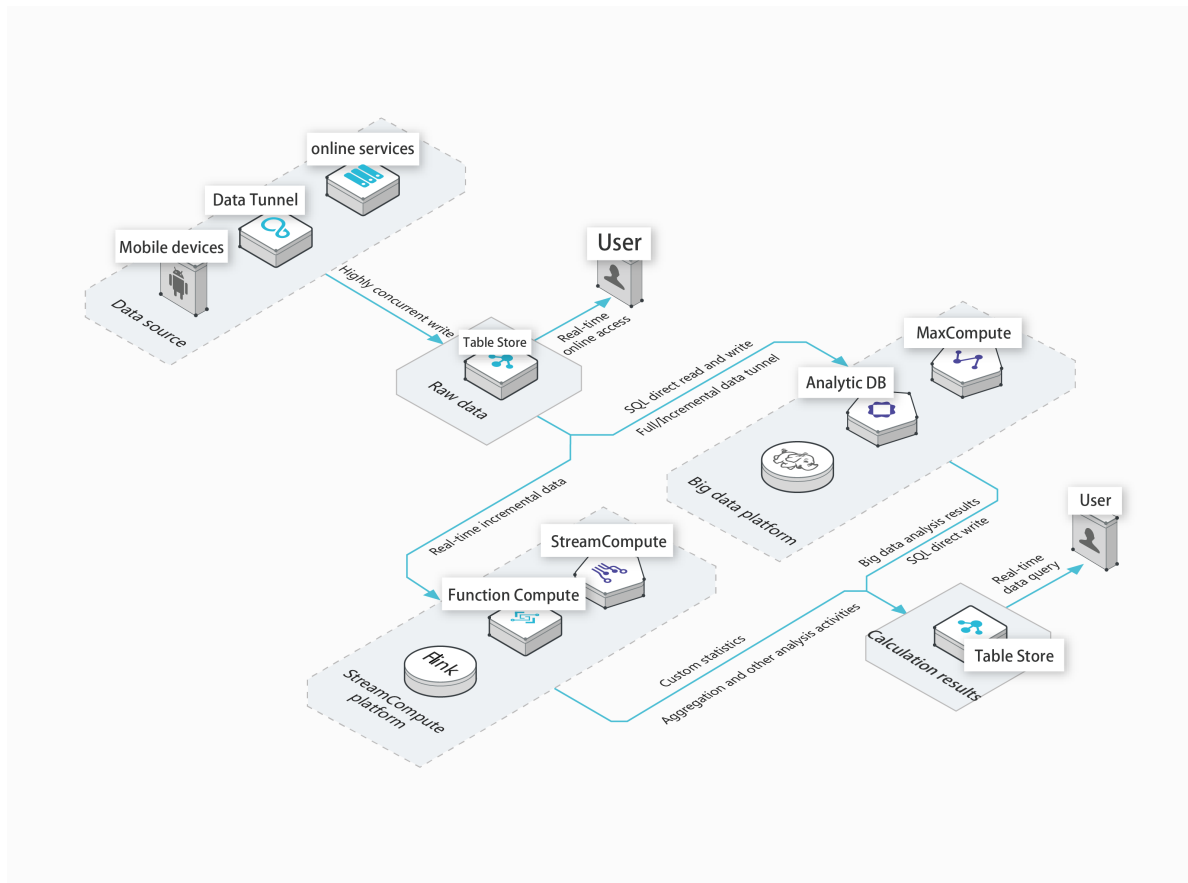
Tablestore can be applied to the following scenarios:

- Scenario 1: Big data storage and analytics

Tablestore provides cost-effective, highly concurrent, and low-latency storage, and online access to large amounts of data. It provides full and incremental data tunnels and supports direct SQL-based read and write operations on various big data analysis platforms such as MaxCompute. An efficient incremental streaming read interface is provided for easy computing of real-time data streams.

Tablestore provides the following features:

- Tablestore supports various big data computing platforms, stream computing services, and real-time computing services.
- Tablestore provides high-performance and capacity instances to meet the requirements of different business.

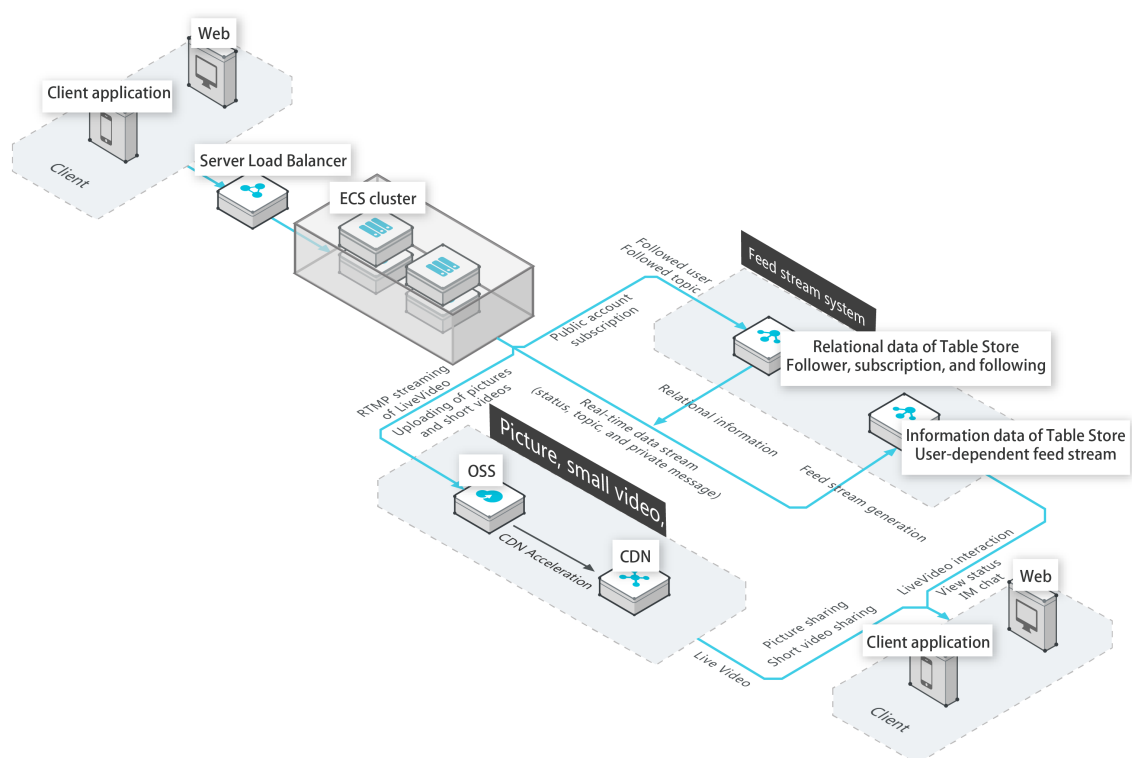


- Scenario 2: Social media feeds on the Internet

You can use Tablestore to store large amounts of instant messaging (IM) messages and social media feed information such as comments, posts, and likes. The elastic resources available for Tablestore can meet application requirements including handling significant traffic fluctuations, high concurrency, and low latency at relatively low costs.

Tablestore provides the following features:

- Built-in auto-increment primary key columns reduce the number of external system dependencies.
- Average read and write performance of high-performance instances are not affected by volumes.
- Highly available storage for large amounts of messages, and multi-terminal message synchronization are supported.

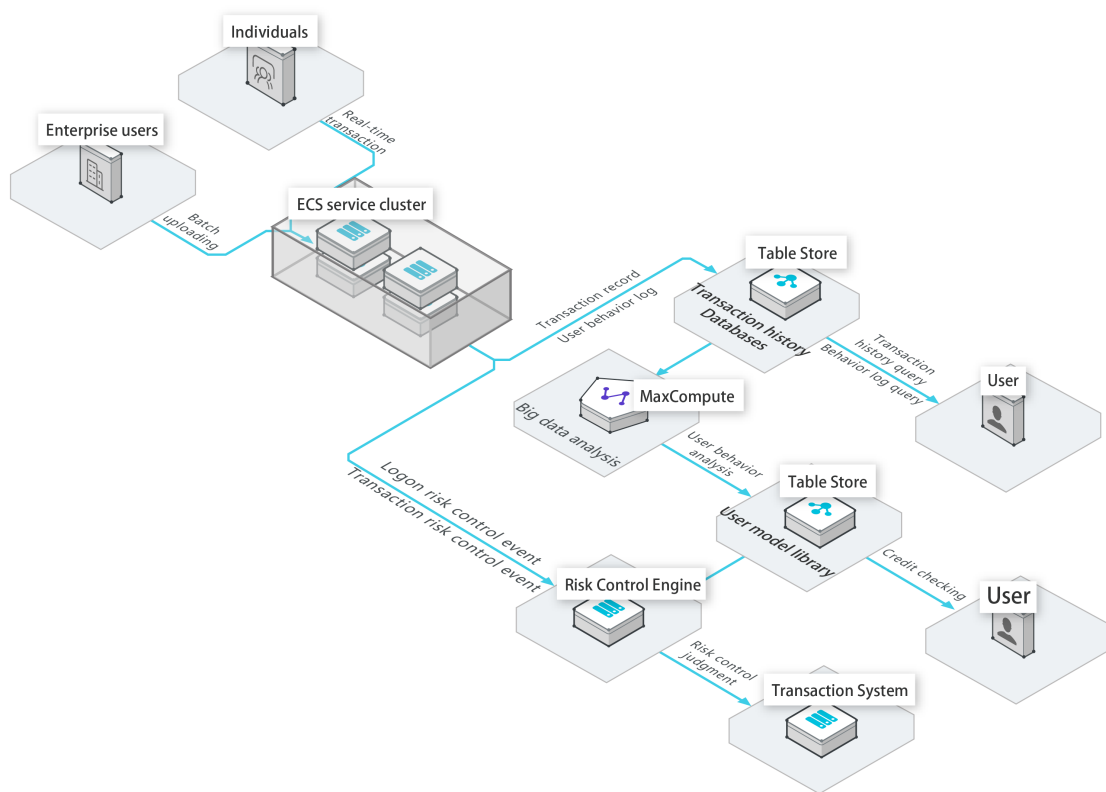


- Scenario 3: Storage and real-time queries of large amounts of transaction records and user models

Tablestore instances are elastic, low latency, and highly concurrent, which provides optimal running conditions for risk control systems. This helps you control transaction risks. Furthermore, the flexible data structure allows your business model to rapidly evolve to meet market demands.

Tablestore provides the following features:

- A table can store full historical transaction records.
- Data is stored in three copies to ensure high consistency and data security.
- The schema-free data model allows you to add attribute columns based on your requirements. This allows rapid service development.

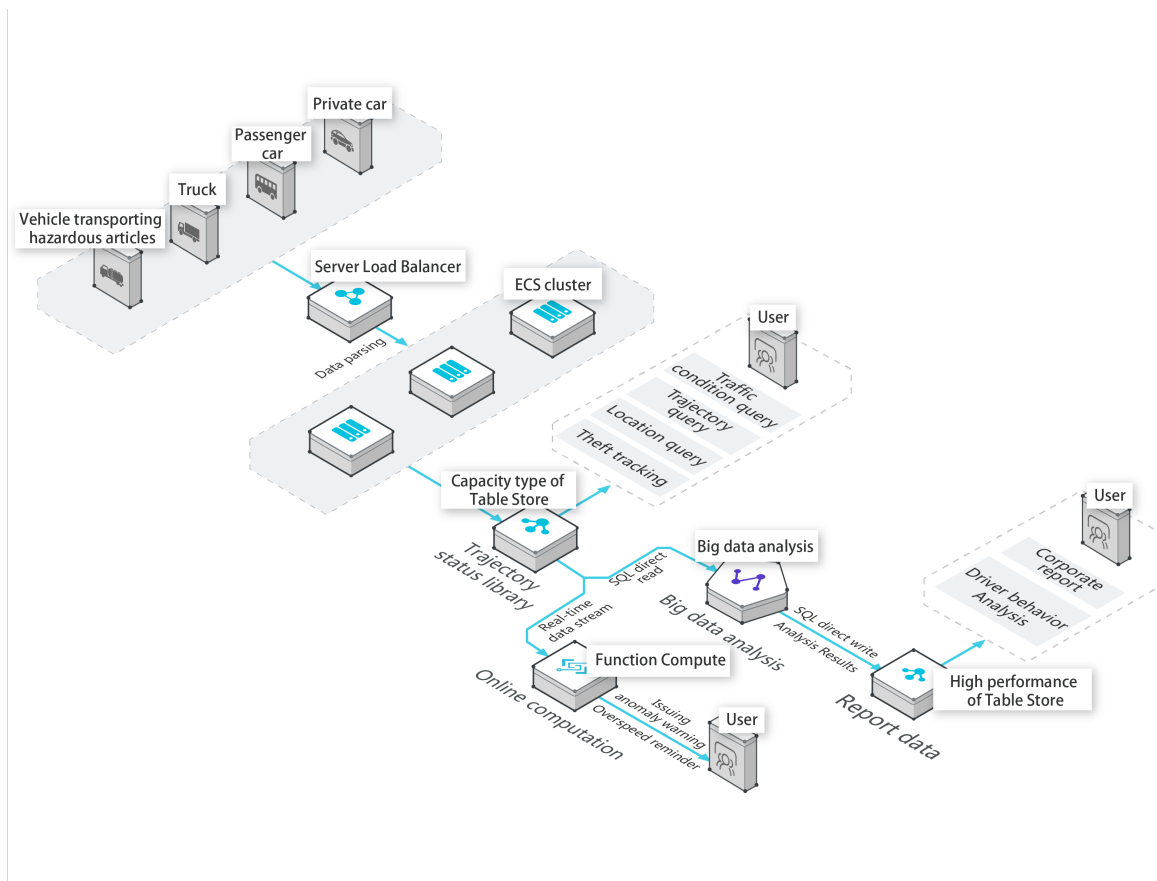


- Scenario 4: Efficient and flexible storage of large amounts of IoT data

The schema-free data model simplifies access to the data collected from different vehicle-mounted devices. Tablestore can be seamlessly integrated with multiple big data analytics platforms and real-time computing services to implement real-time online queries and business report analysis.

Tablestore provides the following features:

- Data is stored in a table without sharding, which simplifies business logic.
- The query performance for vehicle conditions and recommended routes is stable and predictable.
- The schema-free model allows you to store data collected from different vehicle-mounted devices.

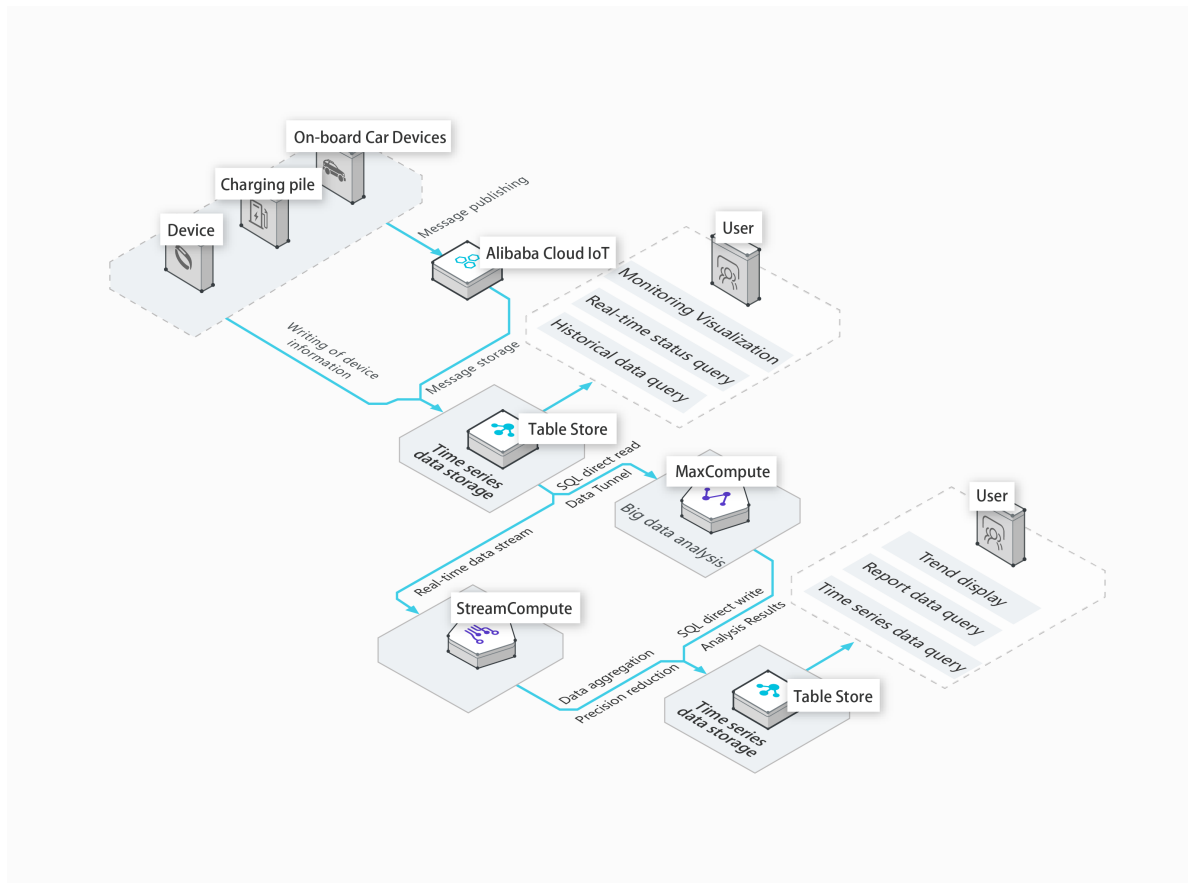


- Scenario 5: Storage of large amounts of IoT data for efficient queries and analysis

Tablestore can be used to store time series data from IoT devices and monitoring systems. It provides API operations to directly read SQL data and incremental data streams, which allow you to implement offline data analysis and real-time stream computing.

Tablestore provides the following features:

- Tablestore can meet the data write and storage requirements of ultra-large-scale IoT devices and monitoring systems.
- Tablestore can integrate with a variety of offline or stream data analysis platforms. This allows you to use a single piece of data for multiple analysis and computing operations.
- Tablestore supports TTL.

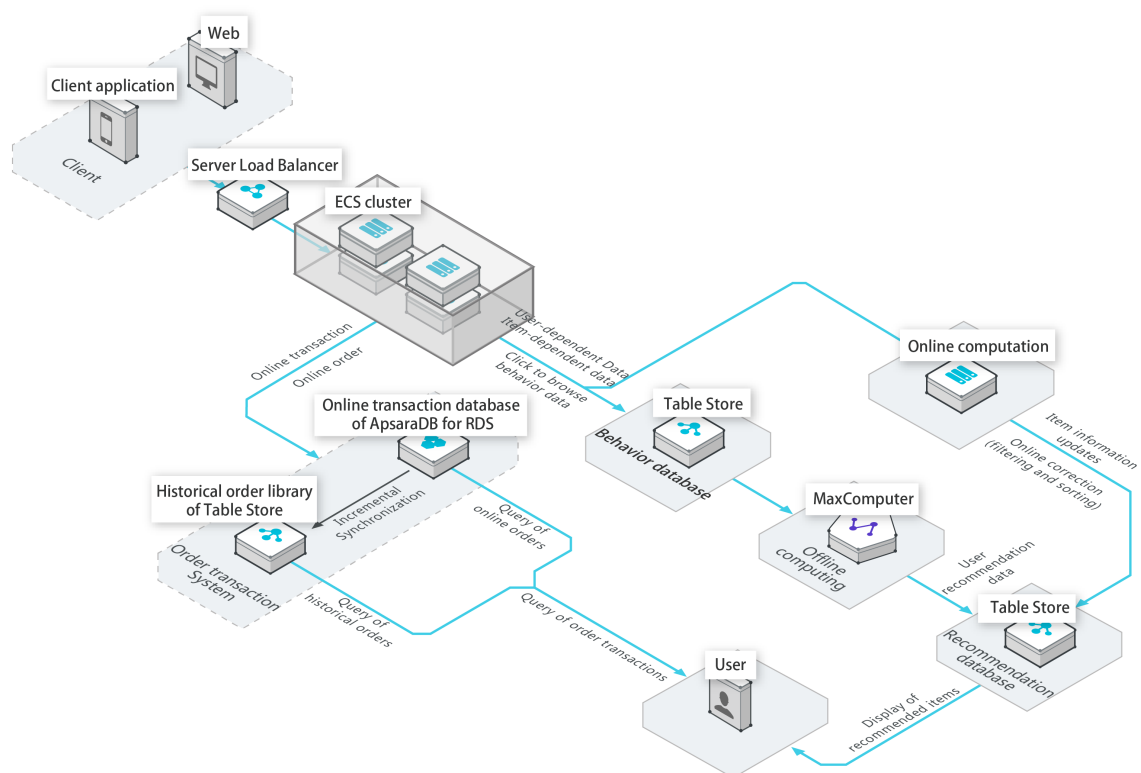


- Scenario 6: Databases for large-scale e-commerce transaction orders and user-specific recommendations

Tablestore can manage large amounts of historical transaction data and improve access performance. Tablestore can be used together with MaxCompute to implement precision marketing and elastic resource storage. This allows you to handle service requests during peak hours when all users go online.

Tablestore provides the following features:

- Resources can be scaled based on data volumes and access concurrency, which allows the service to handle scenarios that feature high access fluctuations during various periods.
- Various big data analytics platforms are supported for direct analysis of user behavior.
- Single-digit millisecond latency for queries on large amounts of data.



7.5. Limits

This topic describes the usage limits of Tablestore.

The following table describes the limits on the usage of Tablestore. A part of limits indicate the maximum values that can be used rather than the suggested values. You can tailor table schemas and row sizes to improve performance.

Item	Limit	Description
The number of instances created in an Apsara Stack tenant account	1024	If you need to increase the maximum number of instances, contact an administrator.
The number of tables in an instance	1024	If you need to increase the maximum number of tables, contact an administrator.
The number of columns in a primary key	1~4	A primary key can contain one to four columns.
The size of the value in a STRING primary key column	1 KB	The size of the value in a STRING primary key column cannot exceed 1 KB.
The size of the value in a STRING attribute column	2 MB	The size of the value in a STRING attribute column cannot exceed 2 MB.
The size of the value in a BINARY primary key column	1 KB	The size of the value in a BINARY primary key column cannot exceed 1 KB.

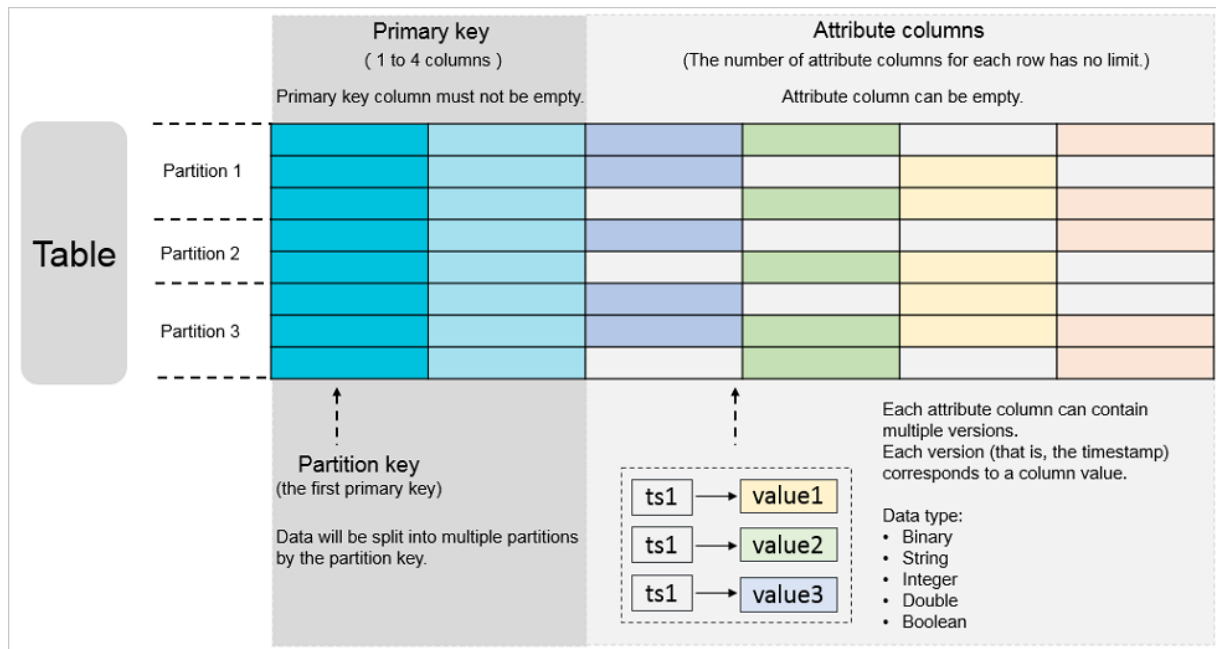
Item	Limit	Description
The size of the value in a BINARY attribute column	2 MB	The size of the value in a BINARY attribute column cannot exceed 2 MB.
The number of attribute columns in a single row	Unlimited	A single row can contain an unlimited number of attribute columns.
The number of attribute columns written by one request	1,024	During a PutRow, UpdateRow, or BatchWriteRow operation, the number of attribute columns written to a single row cannot exceed 1,024.
The data size of a row	Unlimited	The total size of all column names and column values for a row is unlimited.
The number of columns that are specified by the columns_to_get parameter in a read request	0~128	The maximum number of columns obtained from a single row of data in a read request cannot exceed 128.
The number of UpdateTable operations for a table	Upper limit: unlimited Lower limit: unlimited	The frequency of UpdateTable operations for a table is limited.
The frequency of UpdateTable operations for a table	Once every two minutes	The reserved read/write throughput for a table can be adjusted once every two minutes at most.
The number of rows read by one BatchGetRow request	100	None.
The number of rows written by one BatchWriteRow request	200	None.
The size of data written by one BatchWriteRow request	4 MB	None.
Data returned by one GetRange request	5,000 rows or 4 MB	The amount of data returned by a request cannot exceed 5,000 rows or 4 MB. When either of the limits is exceeded, data that exceeds the limits is truncated at the row-level. The data primary key information in the next row is returned.
The data size of an HTTP request body	5 MB	None.

7.6. Terms

This topic describes several basic terms used in Tablestore, including data model, max versions, time to live (TTL), max version offset, primary key and attribute, read/write throughput, region, instance, endpoint, and Serial ATA (SATA).

data model

A data model that consists of tables, rows, primary keys, and attribute columns in Tablestore. The following figure shows an example of a data model.



max versions

A data table attribute that indicates the maximum number of data versions that can be stored in each attribute column of a data table. If the number of versions in an attribute column exceeds the max versions value, the earliest version is asynchronously deleted.

TTL

A data table attribute that indicates the validity period of data in seconds. To save space and reduce costs for data storage, Tablestore deletes any data that exceeds its TTL.

max version offset

A data table attribute that describes the maximum allowable difference between the version to be written and the current time in seconds.

To prevent the writing of unexpected data, a server checks the versions of attribute columns when the server processes writing requests. If the specified version is earlier than the current writing time minus the max version offset value or later than or equal to the current writing time plus the max version offset value, data fails to be written to the row.

The valid version range of an attribute column: **[Data written time - Valid version offset, Data written time + Valid version offset]**. Data written time is the number of seconds that have elapsed since 00:00:00, 1 January 1970. Versions of the attribute columns are written in milliseconds. A version of an attribute column must fall within the valid version range after the version number is converted to seconds (divide by 1,000).

primary key and attribute

A primary key is the unique identifier of each row in a table. A primary key consists of one to four primary key columns. When you create a table, you must define a primary key. You must specify the name, data type, and sequence of each primary key column. The data type of primary key columns can be only STRING, INTEGER, or BINARY. The size of a STRING or BINARY primary key column cannot exceed 1 KB.

An attribute is the attribute data stored in a row. You can create an unlimited number of attribute columns for each row.

read/write throughput

A Tablestore attribute that is measured by read/write capacity units (CUs).

region

An Apsara Stack physical data center. Tablestore is deployed across multiple Apsara Stack regions. Select a region that suits your business requirements.

instance

A logical entity that is used to manage tables in Tablestore. Instances correspond to databases in traditional relational databases. An instance is the basic unit of the Tablestore resource management system. Tablestore allows you to control access and meter resources by instance.

endpoint

The connection URL for each instance. You must specify an endpoint before you perform any operations on Tablestore tables and data.

SATA

A disk that is based on serial connections and provides stronger error-correcting capabilities. Serial ATA aims to improve the reliability of data during transmission.

7.7. Features

7.7.1. Features

This topic describes the basic features of Tablestore, including data partition, load balancing, and automatic recovery from single points of failure (SPOFs).

Tablestore provides the following features:

- Data partition and load balancing

The first column of a primary key in each row of a table is the partition key. The system splits a table into multiple partitions based on the range of partition key values. These partitions are evenly scheduled across different storage nodes. When the data in a partition exceeds the size limit, the partition is automatically split into two smaller partitions. The data and access loads are distributed across these two partitions. The partitions are scheduled to different nodes. As a result, access loads are distributed to different nodes. This allows single-table data and access loads to scale linearly. A partition is a logical organization of data based on the shared storage mechanism. No migration of physical data is involved when a partition is split. However, this may cause the partition to be unable to provide services for 100 milliseconds.

- Automatic recovery from single points of failure (SPOFs)

Each node in the storage engine of Tablestore provides services for multiple data partitions of different tables. The master node manages partition distribution and scheduling, and monitors the health of each service node. If a service node fails, the master node migrates data partitions from the faulty node to other healthy nodes. Services can recover from SPOF in a short time because migrations are performed on the logical level and do not involve the physical migration of data.

7.7.2. Global secondary index

7.7.2.1. Features

This topic describes the features of global secondary index in Tablestore.

Tablestore provides the following features for you to use global secondary index:

- Supports asynchronous data synchronization between a base table and index tables. Under normal network conditions, the data synchronization can reach single-digit millisecond latency.
- Supports single-column indexes and compound indexes.
- Support covered indexes. Predefined columns are specified in advance in a base table. You can create an index table on any predefined column or primary key column of the base table. You can also specify multiple predefined columns of a base table as attribute columns of an index table or choose not to specify attribute columns. If you specify predefined columns of a base table as the attribute columns of an index table, you can directly query this index table instead of querying the base table to obtain the value of the predefined column. For example, a base table includes the primary key columns PK0, PK1, and PK2 and the predefined attribute columns Defined0, Defined1, and Defined2.
 - You can create an index table on PK2 without specifying an attribute column or specifying Defined0 as an attribute column.
 - You can create an index table on PK1 and PK2 without specifying an attribute column or specifying Defined0 as an attribute column.
 - You can create an index table on PK2, PK1, and PK0 and specify Defined0, Defined1, and Defined2 as attribute columns.
 - You can create an index table on Defined0 without specifying an attribute column.
 - You can create an index table on Defined0 and PK1 and specify Defined1 as an attribute column.
 - You can create an index table on Defined1 and Defined0 without specifying an attribute column or specifying Defined2 as an attribute column.
- Supports sparse indexes. You can specify a predefined column in the base table as an attribute column in the index table. A row will be indexed when all indexed columns exist even if the predefined column is excluded from the row of the base table. However, a row will not be indexed when the row excludes one or more indexed columns. For example, a base table includes the primary key columns PK0, PK1, and PK2 and the predefined columns Defined0, Defined1, and Defined2. You can create an index table on Defined0 and Defined1 and specify Defined2 as an attribute column.
 - The index table includes the rows in the base table that include Defined0 and Defined1 but exclude Defined2.
 - The index table excludes the rows in the base table that includes Defined0 and Defined2 but excludes Defined1.
- Supports the deletion or creation of index tables for an existing base table. An index table can contain the existing data of the base table.

- When you query an index table, the query is not performed on the base table. You must query the base table. The automatic query on the base table after a query on an index table will be supported in later versions.

7.7.2.2. Usage notes

This topic describes the terms, limits, and precautions for global secondary indexes.

Terms

Term	Description
index table	The table created based on indexing of columns from the base table. The data in the index table is read-only.
predefined column	The column you predefine when you create a table. Tablestore uses a schema-free model. You can also specify the data type of the column. You can write an unlimited number of columns to a row. You do not need to specify a fixed number of predefined columns in a schema.
single-column index	The index that is created for a single column.
compound index	The index that is created for multiple columns in a table. A compound index can have indexed columns 1 and 2.
indexed attribute column	The predefined column in a base table that is mapped to non-primary key columns in an index table.
autocomplete	Tablestore automatically adds all primary key columns of the base table to the index table.

Limits

- The index table names must be unique in an instance.
- You can create a maximum of five index tables for a base table. If the limit is reached, the index table fails to be created.
- You can create a maximum of 15 predefined columns for a base table. If the limit is reached, the base table fails to be created.
- An index table can contain a maximum of four indexed columns, which are random combinations of the primary keys and predefined columns of the base table. If the limit is reached, the index table fails to be created.
- An index table can contain a maximum of eight attribute columns. If the limit is reached, the index table fails to be created.
- You can set the data type of an indexed column to `STRING`, `INTEGER`, or `BINARY`. The limits on index columns are the same as those on primary key columns of the base table.
- If an index table contains multiple columns, the size limit on the columns is the same as that on primary key columns of the base table.
- If you specify a column of the `STRING` or `BINARY` type as an attribute column of an index table, the limits on attribute columns are the same as those on attribute columns of the base table.
- You cannot create an index table on a table that has the time to live (TTL) parameter configured. If

you want to create index tables on a table that has the TTL parameter configured, use DingTalk to contact technical support.

- You cannot create an index table from a base table that has the max versions parameter configured. If a base table has the max versions parameter configured, index tables fail to be created from the base table. You cannot configure the max versions parameter for a base table that is associated with an index table.
- You cannot customize versions when you write data to a base table that is associated with an index table. Otherwise, the data fails to be written to the base table.
- You cannot use the Stream feature in an index table.
- An indexed base table cannot contain repeated rows that have the same primary key during the same batch write operation. Otherwise, the data fails to be written to the base table.

Usage notes

- Tablestore automatically adds all primary key columns of the base table to the index table. When you scan an index table, you must specify the range of primary key columns. The range can be anywhere from negative infinity to positive infinity. For example, a base table contains the primary key columns PK0 and PK1 and a predefined column Defined0.

When you create an index for the Defined0 column, Tablestore generates an index table that has the primary key columns Defined0, PK0, and PK1. When you create an index for the Defined0 and PK1 columns, Tablestore generates an index table that has the primary key columns Defined0, PK1, and PK0. When you create an index for the PK1 column, Tablestore generates an index table that has the primary key columns PK1 and PK0. When you create an index table, you need only to specify the column that you want to index. Tablestore adds the other primary key columns of the central table to the index table. For example, a base table contains the primary key columns PK0 and PK1 and a predefined column Defined0.

- When you create an index for the Defined0 column, Tablestore generates the index table that has the primary key columns Defined0, PK0, and PK1.
- When you create an index for the PK1 column, Tablestore generates the index table that has the primary key columns PK1 and PK0.
- You can specify predefined columns as attribute columns in the base table. When you specify a predefined column of the base table as an attribute column of the index table, you can search this index table instead of the base table for the column value. However, this increases storage costs. Otherwise, you must query the base table based on the index table. You can choose between these methods.
- We recommend that you do not specify a column whose values are date or time as the first primary key column of an index table because it may slow down index table updates. We recommend that you hash columns related to the time or date and create indexes for the hashed columns. If you have similar requirements, use DingTalk to contact technical support.
- We recommend that you do not define a column of low cardinality or a column that contains enumerated values as the first primary key column of an index table. For example, the gender column restricts the horizontal scalability of the index table and leads to poor write performance.

7.7.2.3. Scenarios

Global secondary index allows you to create an index table based on a specified column. Data in the generated index is sorted by the specified index column. All data written to the base table is synchronized to the index asynchronously. If you only write data to a base table and query index tables created on the table, the query performance can be improved in most scenarios. This topic describes how to use a global secondary index to query phone records.

For example, the following table contains a number of phone records.

CellNumber	StartTime (Unix timestamps)	CalledNumber	Duration	BaseStationNumber
123456	1532574644	654321	60	1
234567	1532574714	765432	10	1
234567	1532574734	123456	20	3
345678	1532574795	123456	5	2
345678	1532574861	123456	100	2
456789	1532584054	345678	200	3

- The `CellNumber` and `StartTime` columns act as the primary key. `CellNumber` represents the caller. `StartTime` represents the call start time.
- The `CalledNumber`, `Duration`, and `BaseStationNumber` columns are predefined columns. `CalledNumber` represents the call recipient. `Duration` represents the call duration. `BaseStationNumber` represents the base station number.

When you end a phone call, information about the call is written to this table. You can create global secondary indexes for different query scenarios. For example, you can create global secondary indexes whose primary key is `CalledNumber` or `BaseStationNumber`.

Assume that you have the following query requirements:

- You want to query the rows where the value of `CellNumber` is `234567`.

Tablestore uses a global ordering model, which sorts all rows by primary key and provides the `getRange` operation to perform sequential scans. When you use `getRange` to scan the base table for this example, you need only to set the minimum and maximum values of PK0 to `234567`, and set the minimum value of PK1 (call start time) to `0` and the maximum value of PK1 to `INT_MAX`.

```

private static void getRangeFromMainTable(SyncClient client, long cellNumber)
{
    RangeRowQueryCriteria rangeRowQueryCriteria = new RangeRowQueryCriteria(TABLE_NAME);
    // Specify the primary key to start from.
    PrimaryKeyBuilder startPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.fromLong(cellNumber));
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(0));
    ;
    rangeRowQueryCriteria.setInclusiveStartPrimaryKey(startPrimaryKeyBuilder.build());
    // Specify the primary key to end with.
    PrimaryKeyBuilder endPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.fromLong(cellNumber));
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.INF_MAX);
    rangeRowQueryCriteria.setExclusiveEndPrimaryKey(endPrimaryKeyBuilder.build());
    rangeRowQueryCriteria.setMaxVersions(1);
    String strNum = String.format("%d", cellNumber);
    System.out.println("The cell number" + strNum + "makes the following calls:");
    while (true) {
        GetRangeResponse getRangeResponse = client.getRange(new GetRangeRequest(rangeRowQueryCriteria));
        for (Row row : getRangeResponse.getRows()) {
            System.out.println(row);
        }
        // If the nextStartPrimaryKey value is not null, continue the read operation.
        if (getRangeResponse.getNextStartPrimaryKey() != null) {
            rangeRowQueryCriteria.setInclusiveStartPrimaryKey(getRangeResponse.getNextStartPrimaryKey());
        } else {
            break;
        }
    }
}

```

- You want to query the rows where the value of CalledNumber is 123456 .

Tablestore sorts all rows based on primary keys. Queries that involve this column are slow and inefficient because CalledNumber is a predefined column. Therefore, you create an index table based on CalledNumber to improve query speed and efficiency.

IndexOnBeCalledNumber :

PK0	PK1	PK2
CalledNumber	CellNumber	StartTime
123456	234567	1532574734
123456	345678	1532574795
123456	345678	1532574861
654321	123456	1532574644

PK0	PK1	PK2
765432	234567	1532574714
345678	456789	1532584054

 **Note** Tablestore automatically adds all primary key columns of the central table to the index table. The primary key of the global secondary index consists of the index column and the primary key columns of the base table. Therefore, the global secondary index contains three primary key columns.

CalledNumber is a primary key column of `IndexOnBeCalledNumber`. You can perform a query on this index table to query the rows where the value of CalledNumber is 123456.

```
private static void getRangeFromIndexTable(SyncClient client, long cellNumber) {
    RangeRowQueryCriteria rangeRowQueryCriteria = new RangeRowQueryCriteria(INDEX0_NAME);
    // Specify the primary key to start from.
    PrimaryKeyBuilder startPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
    startPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_1, PrimaryKeyValue.fromLong(cellNumber));
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MIN);
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.INF_MIN);
    rangeRowQueryCriteria.setInclusiveStartPrimaryKey(startPrimaryKeyBuilder.build());
    // Specify the primary key to end with.
    PrimaryKeyBuilder endPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
    endPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_1, PrimaryKeyValue.fromLong(cellNumber));
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MAX);
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.INF_MAX);
    rangeRowQueryCriteria.setExclusiveEndPrimaryKey(endPrimaryKeyBuilder.build());
    rangeRowQueryCriteria.setMaxVersions(1);
    String strNum = String.format("%d", cellNumber);
    System.out.println("The cell number" + strNum + "was called by the following numbers:");
    while (true) {
        GetRangeResponse getRangeResponse = client.getRange(new GetRangeRequest(rangeRowQueryCriteria));
        for (Row row : getRangeResponse.getRows()) {
            System.out.println(row);
        }
        // If the nextStartPrimaryKey value is not null, continue the read operation.
        if (getRangeResponse.getNextStartPrimaryKey() != null) {
            rangeRowQueryCriteria.setInclusiveStartPrimaryKey(getRangeResponse.getNextStartPrimaryKey());
        } else {
            break;
        }
    }
}
```

- You want to query the rows where the value of BaseStationNumber is `002` and the value of StartTime is `1532574740`.

This query specifies `BaseStationNumber` and `StartTime` as conditions. Therefore, you can create a compound index based on the `BaseStationNumber` and `StartTime` columns.

`IndexOnBaseStation1` :

PK0	PK1	PK2
BaseStationNumber	StartTime	CellNumber
001	1532574644	123456
001	1532574714	234567
002	1532574795	345678
002	1532574861	345678
003	1532574734	234567
003	1532584054	456789

The following code provides an example on how to query the `IndexOnBaseStation1` index table:

```

private static void getRangeFromIndexTable(SyncClient client,
                                           long baseStationNumber,
                                           long startTime) {
    RangeRowQueryCriteria rangeRowQueryCriteria = new RangeRowQueryCriteria(INDEX1_NAME);
    // Specify the primary key to start from.
    PrimaryKeyBuilder startPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
    startPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber));
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(startTime));
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MIN);
    rangeRowQueryCriteria.setInclusiveStartPrimaryKey(startPrimaryKeyBuilder.build());
    // Specify the primary key to end with.
    PrimaryKeyBuilder endPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
    endPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber));
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.INF_MAX);
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MAX);
    rangeRowQueryCriteria.setExclusiveEndPrimaryKey(endPrimaryKeyBuilder.build());
    rangeRowQueryCriteria.setMaxVersions(1);
    String strBaseStationNum = String.format("%d", baseStationNumber);
    String strStartTime = String.format("%d", startTime);
    System.out.println("All called numbers forwarded by the base station" + strBaseStationNum + "that start from" + strStartTime + "are listed:");
    while (true) {
        GetRangeResponse getRangeResponse = client.getRange(new GetRangeRequest(rangeRowQueryCriteria));
        for (Row row : getRangeResponse.getRows()) {
            System.out.println(row);
        }
        // If the nextStartPrimaryKey value is not null, continue the read operation.
        if (getRangeResponse.getNextStartPrimaryKey() != null) {
            rangeRowQueryCriteria.setInclusiveStartPrimaryKey(getRangeResponse.getNextStartPrimaryKey());
        } else {
            break;
        }
    }
}

```

- You want to query the rows where the value of BaseStationNumber is 003 and the value of StartTime ranges from 1532574861 to 1532584054 and return only the Duration column.

In this query, you specify both BaseStationNumber and StartTime as conditions, but only the Duration column is returned. You can initiate a query on the previous index table, and then query Duration by querying the base table.

```

private static void getRowFromIndexAndMainTable(SyncClient client,
                                                long baseStationNumber,
                                                long startTime,
                                                long endTime,
                                                String colName) {
    RangeRowQueryCriteria rangeRowQueryCriteria = new RangeRowQueryCriteria(INDEX1_NAME);
    // Specify the primary key to start from.
    PrimaryKeyBuilder startPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
    startPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber));
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(startTime));
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MIN);
    rangeRowQueryCriteria.setInclusiveStartPrimaryKey(startPrimaryKeyBuilder.build());
    // Specify the primary key to end with.
    PrimaryKeyBuilder endPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
    endPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber));
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(endTime));
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MAX);
    rangeRowQueryCriteria.setExclusiveEndPrimaryKey(endPrimaryKeyBuilder.build());
    rangeRowQueryCriteria.setMaxVersions(1);
    String strBaseStationNum = String.format("%d", baseStationNumber);
    String strStartTime = String.format("%d", startTime);
    String strEndTime = String.format("%d", endTime);
    System.out.println("All called numbers forwarded by the base station" + strBaseStationNum + "that start from" + strStartTime + "and end at" + strEndTime + "are listed:");
    while (true) {
        GetRangeResponse getRangeResponse = client.getRange(new GetRangeRequest(rangeRowQueryCriteria));
        for (Row row : getRangeResponse.getRows()) {
            System.out.println(row);
        }
        // If the nextStartPrimaryKey value is not null, continue the read operation.
        if (getRangeResponse.getNextStartPrimaryKey() != null) {
            rangeRowQueryCriteria.setInclusiveStartPrimaryKey(getRangeResponse.getNextStartPrimaryKey());
        } else {
            break;
        }
    }
}

```

```

    PrimaryKeyBuilder startPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
    startPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber));
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(startTime));
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MIN);
    rangeRowQueryCriteria.setInclusiveStartPrimaryKey(startPrimaryKeyBuilder.build());
    // Specify the primary key to end with.
    PrimaryKeyBuilder endPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
    endPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber));
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(endTime));
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MAX);
    rangeRowQueryCriteria.setExclusiveEndPrimaryKey(endPrimaryKeyBuilder.build());
    rangeRowQueryCriteria.setMaxVersions(1);
    String strBaseStationNum = String.format("%d", baseStationNumber);
    String strStartTime = String.format("%d", startTime);
    String strEndTime = String.format("%d", endTime);
    System.out.println("The duration of calls forwarded by the base station" + strBaseStationNum + "from"
    + strStartTime + "to" + strEndTime + "is listed:");
    while (true) {
        GetRangeResponse getRangeResponse = client.getRange(new GetRangeRequest(rangeRowQueryCriteria));
        for (Row row : getRangeResponse.getRows()) {
            PrimaryKey curIndexPrimaryKey = row.getPrimaryKey();
            PrimaryKeyColumn mainCalledNumber = curIndexPrimaryKey.getPrimaryKeyColumn(PRIMARY_KEY_NAME_1);
            PrimaryKeyColumn callStartTime = curIndexPrimaryKey.getPrimaryKeyColumn(PRIMARY_KEY_NAME_2);
            PrimaryKeyBuilder mainTablePKBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
            mainTablePKBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, mainCalledNumber.getValue());
            ;
            mainTablePKBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, callStartTime.getValue());
            PrimaryKey mainTablePK = mainTablePKBuilder.build(); // Specify primary keys for the base table.
            // Query the base table.
            SingleRowQueryCriteria criteria = new SingleRowQueryCriteria(TABLE_NAME, mainTablePK);
            criteria.addColumnsToGet(colName); // Read the Duration column value of the base table.
            // Set the latest version to read.
            criteria.setMaxVersions(1);
            GetRowResponse getRowResponse = client.getRow(new GetRowRequest(criteria));
            Row mainTableRow = getRowResponse.getRow();
            System.out.println(mainTableRow);
        }
        // If the nextStartPrimaryKey value is not null, continue the read operation.
        if (getRangeResponse.getNextStartPrimaryKey() != null) {
            rangeRowQueryCriteria.setInclusiveStartPrimaryKey(getRangeResponse.getNextStartPrimaryKey());
        } else {
            break;
        }
    }
}

```


To improve query performance, you can create a compound index based on **BaseStationNumber** and **StartTime** and specify **Duration** as an attribute column of the index table.

The following index table is created.

IndexOnBaseStation2 :

PK0	PK1	PK2	Defined0
BaseStationNumber	StartTime	CellNumber	Duration
001	1532574644	123456	60
001	1532574714	234567	10
002	1532574795	345678	5
002	1532574861	345678	100
003	1532574734	234567	20
003	1532584054	456789	200

The following code provides an example on how to query the **IndexOnBaseStation2** index table:

```

private static void getRangeFromIndexTable(SyncClient client,
                                           long baseStationNumber,
                                           long startTime,
                                           long endTime,
                                           String colName) {
    RangeRowQueryCriteria rangeRowQueryCriteria = new RangeRowQueryCriteria(INDEX2_NAME);
    // Specify the primary key to start from.
    PrimaryKeyBuilder startPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
    startPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber));
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(startTime));
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MIN);
    rangeRowQueryCriteria.setInclusiveStartPrimaryKey(startPrimaryKeyBuilder.build());
    // Specify the primary key to end with.
    PrimaryKeyBuilder endPrimaryKeyBuilder = PrimaryKeyBuilder.createPrimaryKeyBuilder();
    endPrimaryKeyBuilder.addPrimaryKeyColumn(DEFINED_COL_NAME_3, PrimaryKeyValue.fromLong(baseStationNumber));
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(endTime));
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_1, PrimaryKeyValue.INF_MAX);
    rangeRowQueryCriteria.setExclusiveEndPrimaryKey(endPrimaryKeyBuilder.build());
    // Specify the name of the column to read.
    rangeRowQueryCriteria.addColumnsToGet(colName);
    rangeRowQueryCriteria.setMaxVersions(1);
    String strBaseStationNum = String.format("%d", baseStationNumber);
    String strStartTime = String.format("%d", startTime);
    String strEndTime = String.format("%d", endTime);
    System.out.println("The duration of calls forwarded by the base station" + strBaseStationNum + "from " + strStartTime + "to" + strEndTime + "is listed:");
    while (true) {
        GetRangeResponse getRangeResponse = client.getRange(new GetRangeRequest(rangeRowQueryCriteria));
        for (Row row : getRangeResponse.getRows()) {
            System.out.println(row);
        }
        // If the nextStartPrimaryKey value is not null, continue the read operation.
        if (getRangeResponse.getNextStartPrimaryKey() != null) {
            rangeRowQueryCriteria.setInclusiveStartPrimaryKey(getRangeResponse.getNextStartPrimaryKey());
        } else {
            break;
        }
    }
}

```

If you do not specify `Duration` as an attribute column for an index table, you must retrieve `Duration` by querying the base table. However, when you specify `Duration` as an attribute column for an index table, this column is stored in both the base table and the index table. The configuration improves query performance at the cost of storage space consumption.

- You want to query the total call duration, average call duration, maximum call duration, and minimum

call duration of all calls forwarded by the base station 003 and whose call start time range from 1532574861 to 1532584054 .

In this query, you want to query the statistics for the duration of all phone calls instead of the duration of each call that is queried in the previous scenario. You can obtain results by using the same method as in the previous query. Then, you can perform calculations on the Duration column to obtain the required result. You can also use SQL-on-OTS to directly return the final statistical results without the need for client computing. You can use most MySQL syntax in SQL-on-OTS. Additionally, SQL-on-OTS enable you to process complicated calculations that are applicable to your business.

8. ApsaraDB RDS

8.1. What is ApsaraDB RDS?

ApsaraDB Relational Database Service (RDS) is a stable, reliable, and scalable online database service. Based on the distributed file system and high-performance storage, ApsaraDB RDS provides a set of solutions for disaster recovery, backup, restoration, monitoring, and migration.

ApsaraDB RDS supports four database engines, which are MySQL, SQL Server, PolarDB, and PostgreSQL. You can create database instances based on these engines to meet your business requirements.

ApsaraDB RDS for MySQL

Originally based on a branch of MySQL, ApsaraDB RDS for MySQL provides excellent performance. It is a tried and tested solution that handled the high-volume concurrent traffic during Double 11. ApsaraDB RDS for MySQL provides basic features such as whitelist configuration, backup and restoration, Transparent Data Encryption (TDE), data migration, and management for instances, accounts, and databases. ApsaraDB RDS for MySQL also provides the following advanced features:

- **Read-only instance:** In scenarios where ApsaraDB RDS for MySQL handles a small number of write requests but a large number of read requests, you can create read-only instances to scale up the reading capability and increase the application throughput.
- **Read/write splitting:** The read/write splitting feature provides a read/write splitting endpoint. This endpoint enables an automatic link for the primary instance and all its read-only instances. An application can connect to the read/write splitting endpoint to read and write data. Write requests are distributed to the primary instance and read requests are distributed to read-only instances based on their weights. To scale up the reading capability of the system, you can add more read-only instances.

ApsaraDB RDS for SQL Server

ApsaraDB RDS for SQL Server provides strong support for a variety of enterprise applications under the high-availability architecture. ApsaraDB RDS for SQL Server can also restore data to a specific point in time, which reduces costs.

ApsaraDB RDS for SQL Server provides basic features such as whitelist configuration, backup and restoration, TDE, data migration, and management for instances, accounts, and databases.

PolarDB

PolarDB is a stable, secure, and scalable enterprise-class relational database service. Based on PostgreSQL, PolarDB enhances performance, application solutions, and compatibility. PolarDB also provides the capability of directly running Oracle applications. You can run a variety of enterprise applications on PolarDB stably at low costs.

PolarDB provides features such as account management, resource monitoring, backup and restoration, and security control. These features are under continuous improvement, and more features are under development to adapt to PolarDB.

ApsaraDB RDS for PostgreSQL

ApsaraDB RDS for PostgreSQL is an advanced open source database service that is fully compatible with SQL and supports a diverse range of data formats such as JSON, IP, and geometric data. In addition to support for features such as transactions, subqueries, multi-version concurrency control (MVCC), and data integrity check, ApsaraDB RDS for PostgreSQL integrates a series of features including high availability, backup, and restoration to ease operations and maintenance loads.

8.2. Benefits

8.2.1. Ease of use

ApsaraDB RDS is a ready-to-use service that provides features such as on-demand upgrade, easy management, high transparency, and high compatibility.

Ready-to-use

You can use API operations to create ApsaraDB RDS instances of your desired instance type.

On-demand upgrade

When the database load or data volume changes, you can upgrade an ApsaraDB RDS instance by changing its instance type. The upgrades do not interrupt the data link service.

Transparency and compatibility

You can easily use ApsaraDB RDS in the same way as native database engines without the need to acquire new knowledge. ApsaraDB RDS is compatible with your existing programs and tools. Data can be migrated to ApsaraDB RDS by using ordinary import and export tools.

Easy management

You can add, delete, restart, back up, and restore databases by using the Apsara Uni-manager Management Console.

8.2.2. High performance

ApsaraDB RDS provides parameter optimization, SQL optimization, and high-end backend hardware to implement high performance.

Parameter optimization

All the parameters of ApsaraDB RDS instances are optimized based on years of production. Professional database administrators continue to optimize ApsaraDB RDS instances over their lifecycles to ensure that ApsaraDB RDS runs at peak efficiency.

SQL optimization

ApsaraDB RDS locks inefficient SQL statements and provides recommendations to optimize code.

8.2.3. High security

ApsaraDB RDS implements DDoS attack prevention, access control, system security, and Transparent Data Encryption (TDE) to ensure the security of databases.

DDoS attack prevention

 **Note** You must activate Apsara Stack security services to use this feature.

When you access an ApsaraDB RDS instance from the Internet, the instance is vulnerable to DDoS attacks. When a DDoS attack is detected, the ApsaraDB RDS security system first scrubs the inbound traffic. If traffic scrubbing is not sufficient or if the blackhole triggering threshold is reached, blackhole filtering is triggered.

Access control

You can configure an IP address whitelist for ApsaraDB RDS to allow access for specified IP addresses and deny access for all others.

Each account can view and manage only their own respective databases.

System security

ApsaraDB RDS is protected by several layers of firewalls capable of blocking a variety of attacks to ensure data security.

You cannot directly log on to ApsaraDB RDS servers. Only the ports required for specific database services are provided.

ApsaraDB RDS servers cannot initiate an external connection. They can only receive access requests.

TDE

TDE can be used to perform real-time I/O encryption and decryption on instance data files. Data is encrypted before it is written to disks and decrypted before it is read from disks to the memory. TDE does not increase the size of data files. Developers do not need to modify their applications before they use the TDE feature.

8.2.4. High reliability

ApsaraDB RDS provides hot standby, multi-copy redundancy, data backup, and data restoration to implement high reliability.

Hot standby

ApsaraDB RDS adopts a hot standby architecture. If the primary server fails, services fail over to the secondary server within seconds. Applications that run on the servers are not affected by the failover process and can continue to run normally.

Multi-copy redundancy

ApsaraDB RDS servers implement a Redundant Array of Independent Disks (RAID) architecture to store data. Data backup files are stored on Object Storage Service (OSS).

Data backup

ApsaraDB RDS provides an automatic backup mechanism. You can select a time range to perform backups or initiate temporary backups to meet your business requirements.

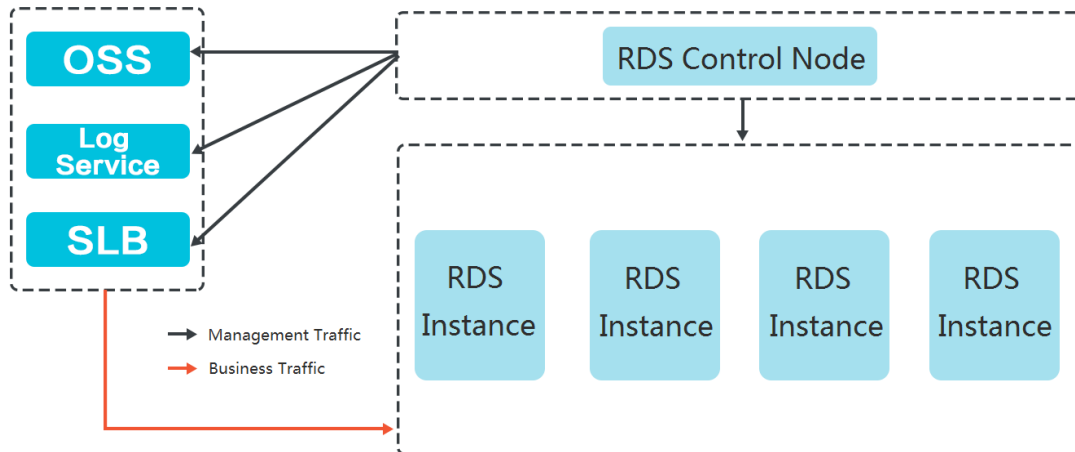
Data restoration

Data can be restored from backup sets or previous points in time by using cloned instances. After data is verified, the data can be migrated back to the primary ApsaraDB RDS instance.

8.3. Architecture

The following figure shows the system architecture of ApsaraDB RDS.

ApsaraDB RDS system architecture



8.4. Features

8.4.1. Data link service

ApsaraDB RDS provides all data link services, including Domain Name System (DNS), and Server Load Balancer (SLB).

ApsaraDB RDS uses native database engines that have similar database operations to minimize learning costs and facilitate database access.

DNS

The DNS module can dynamically resolve domain names to IP addresses. Therefore, IP address changes do not affect the performance of ApsaraDB RDS instances. After the domain name of an ApsaraDB RDS instance is configured in the connection pool, the ApsaraDB RDS instance can be accessed even if its corresponding IP address changes.

For example, assume that the domain name of an ApsaraDB RDS instance is test.rds.aliyun.com, and its corresponding IP address is 10.10.10.1. The instance can be accessed when test.rds.aliyun.com or 10.10.10.1 is configured in the connection pool of a program.

After this ApsaraDB RDS instance is migrated or its version is upgraded, the IP address may change to 10.10.10.2. If the domain name test.rds.aliyun.com is configured in the connection pool, the instance can still be accessed. However, if the IP address 10.10.10.1 is configured in the connection pool, the instance is no longer accessible.

SLB

The SLB module provides both the internal and public IP addresses of an ApsaraDB RDS instance. Therefore, server changes do not affect the performance of the instance.

For example, assume that the internal IP address of an ApsaraDB RDS instance is 10.1.1.1, and the corresponding Proxy or DB Engine runs on 192.168.0.1. The SLB module typically redirects all traffic destined for 10.1.1.1 to 192.168.0.1. If 192.168.0.1 fails, another server in the hot standby state with the IP address 192.168.0.2 takes over for the initial server. In this case, the SLB module redirects all traffic destined for 10.1.1.1 to 192.168.0.2, and the ApsaraDB RDS instance continues to provide services normally.

8.4.2. High-availability service

The high-availability (HA) service consists of modules such as Detection, Repair, and Notice, as well as multiple HA policies.

The HA service ensures the availability of data link services and processes internal database exceptions.

Detection

The Detection module checks whether the primary and secondary nodes of the DB Engine are providing services normally. The HA node uses heartbeat information taken at 8 to 10 second intervals to determine the health status of the primary node. This information, along with the health status of the secondary node and heartbeat information from other HA nodes, provides a reference for the Detection module. All this information helps the module avoid misjudgment caused by exceptions such as network jitter. Failover can be completed within a short time.

Repair

The Repair module maintains the replication relationship between the primary and secondary nodes of the DB Engine. It can also correct errors that occur on the nodes during daily operations.

For example:

- It can automatically restore primary/secondary replication after a disconnection.
- It can automatically repair table-level damage to the primary or secondary node.
- It can save and automatically repair the primary or secondary node when the node fails.

Notice

The Notice module informs the Server Load Balancer (SLB) or Proxy module of status changes to the primary and secondary nodes to ensure that you always access the correct node.

For example, the Detection module discovers problems with the primary node and instructs the Repair module to resolve these problems. If the Repair module fails to resolve a problem, it instructs the Notice module to perform traffic switchover. The Notice module forwards the switching request to the SLB or Proxy module. Then, all traffic is redirected to the secondary node. Meanwhile, the Repair module creates a new secondary node on a different physical server and synchronizes this change back to the Detection module. The Detection module rechecks the health status of the instance.

HA policies

Each HA policy defines a combination of service priorities and data replication modes to meet the needs of your business.

The following service priorities are available:

- Recovery time objective (RTO): The database preferentially restores services to maximize the availability time. Use the RTO policy if you require longer database uptime.
- Recovery point objective (RPO): The database preferentially ensures data reliability to minimize data

loss. Use the RPO policy if you require high data consistency.

The following data replication modes are available:

- **Asynchronous replication (Async):** When an application initiates an update request such as add, delete, or modify operations, the primary node responds to the application immediately after the primary node completes the operation. Then, the primary node asynchronously replicates data to the secondary node. This ensures that the operation of the primary database is not affected if the secondary node is unavailable. Data inconsistencies may occur if the primary node is unavailable.
- **Forced synchronous replication (Sync):** When an application initiates an update request such as add, delete, or modify operations, the primary node replicates data to the secondary node immediately after the primary node completes the operation. The primary node then waits for the secondary node to return a success message before the primary node responds to the application. The primary node synchronously replicates data to the secondary node. Therefore, the operation of the primary node is affected if the secondary node is unavailable. Data remains consistent even when the primary node is unavailable.
- **Semi-synchronous replication (Semi-Sync):** Data is typically replicated in Sync mode. When the primary node is replicating data to the secondary node, if an exception occurs and causes the primary and secondary nodes to be unable to communicate with each other, the primary node suspends response to the application. If the connection cannot be restored, the primary node degrades to the Async mode and restores response to the application after the Sync replication times out. In such a situation, data may be inconsistent if the primary node becomes unavailable. After the secondary node or network connection is restored, data replication between the two nodes is resumed, and the data replication mode changes from Async to Sync.

You can select different combinations of service priorities and data replication modes to improve availability based on your business.

8.4.3. Backup and recovery service

This service supports data backup, storage, and recovery features.

ApsaraDB RDS can back up databases anytime and restore them to a point in time based on backup policies, which makes data more traceable.

Backup

The Backup module compresses and uploads data and logs on both the primary and secondary nodes. By default, ApsaraDB RDS uploads backup files to Object Storage Service (OSS). When the secondary node operates normally, backups are always created on the secondary node. This way, the services on the primary node are not affected. When the secondary node is unavailable or damaged, the Backup module creates backups on the primary node.

Recovery

The Recovery module restores backup files from OSS to a destination node.

- **Primary node rollback:** rolls back the primary node to a specific point in time when an operation error occurs.
- **Secondary node repair:** creates a new secondary node to reduce risks when an irreparable fault occurs on the secondary node.
- **Read-only instance creation:** creates a read-only instance from backup files.

Storage

The Storage module uploads, dumps, and downloads backup files. All backup data is uploaded to OSS for storage. You can obtain temporary links to download the data. In specific scenarios, the Storage module allows you to dump backup files from OSS to Archive Storage for more cost-effective and longer-term offline storage.

8.4.4. Monitoring service

ApsaraDB RDS provides multilevel monitoring services across the physical, network, and application layers to ensure service availability.

Service

The Service module tracks the status of services. For example, the Service module monitors whether other cloud services on which ApsaraDB RDS depends are operating normally, such as Server Load Balancer (SLB) and Object Storage Service (OSS). The monitored metrics include features and response time. The Service module also uses logs to determine whether the internal services of ApsaraDB RDS are operating properly.

Network

The Network module tracks statuses at the network layer. The module monitors the connectivity between Elastic Compute Service (ECS) and ApsaraDB RDS and between physical servers of ApsaraDB RDS. It also monitors the rates of packet loss on vRouters and vSwitches.

OS

The OS module tracks the statuses of hardware and OS kernel. The following metrics are monitored:

- Hardware maintenance: The OS module constantly checks the operating status of the CPU, memory, motherboard, and storage device. It can predict faults and automatically submit repair reports when it determines a fault is likely to occur.
- OS kernel monitoring: The OS module tracks all database calls and analyzes the causes of slow calls or call errors based on the kernel status.

Instance

The Instance module collects the following information about ApsaraDB RDS instances:

- Instance availability information
- Instance capacity and performance metrics
- Instance SQL execution records

8.4.5. Scheduling service

The Resource module implements the scheduling of resources and services.

Resource

The Resource module allocates and integrates underlying ApsaraDB RDS resources when you enable and migrate instances. When you create an instance by using the ApsaraDB RDS console or an API operation, the Resource module calculates the most suitable host to carry traffic to and from the instance. This module also allocates and integrates the underlying resources required to migrate RDS instances. After instances are repeatedly created, deleted, or migrated, the Resource module calculates the degree of resource fragmentation. In addition, it integrates resources on a regular basis to improve the service carrying capacity.

8.4.6. Migration service

ApsaraDB RDS provides Data Transmission Service (DTS) to help you migrate databases.

The migration service helps you migrate data from on-premises databases to ApsaraDB RDS instances or between ApsaraDB RDS instances.

DTS

DTS enables data migration from on-premises databases to ApsaraDB RDS instances or between ApsaraDB RDS instances.

DTS provides three migration methods: schema migration, full migration, and incremental migration.

- Schema migration

DTS migrates the schema definitions of migration objects to the destination instance. Tables, views, triggers, stored procedures, and stored functions can be migrated in this mode.

- Full migration

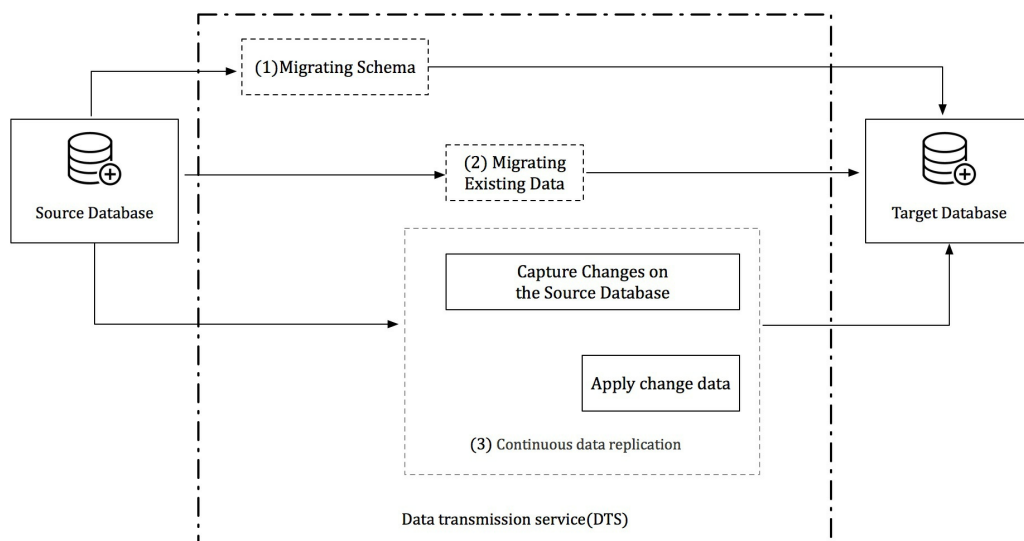
DTS migrates all data of migration objects from the source database to the destination instance.

 **Notice** For data consistency purposes, non-transaction tables that do not have primary keys are locked during a full migration. You cannot write data to locked tables. The lock duration is determined by the data volume in the tables. The tables are unlocked only after they are fully migrated.

- Incremental migration

DTS synchronizes data changes made in the migration process to the destination instance.

 **Notice** If a DDL operation is performed when data is migrated, schema changes are not synchronized to the destination instance.



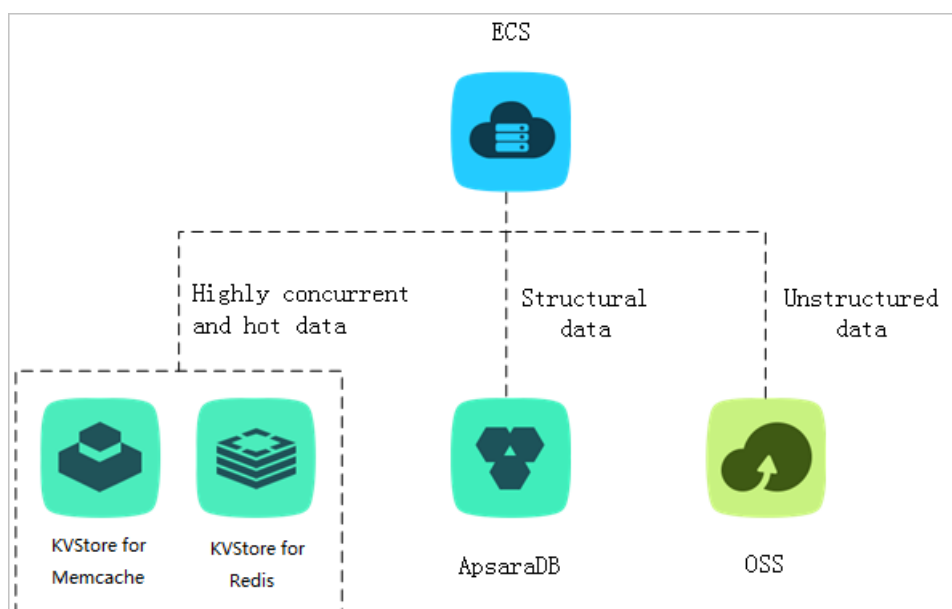
8.5. Scenarios

8.5.1. Diversified data storage

ApsaraDB RDS provides cache data persistence and multi-structure data storage.

You can diversify the storage capabilities of ApsaraDB RDS by using services such as KVStore for Redis and Object Storage Service (OSS), as shown in the [Diversified data storage](#) figure.

Diversified data storage



Cache data persistence

ApsaraDB RDS can be used in conjunction with KVStore for Memcache and KVStore for Redis to form a high-throughput and low-latency storage solution. These cache services have the following benefits over ApsaraDB RDS:

- High response speed: The request latency of KVStore for Memcache and KVStore for Redis is only a few milliseconds.
- The cache area supports a higher number of queries per second (QPS) than ApsaraDB RDS.

Multi-structure data storage

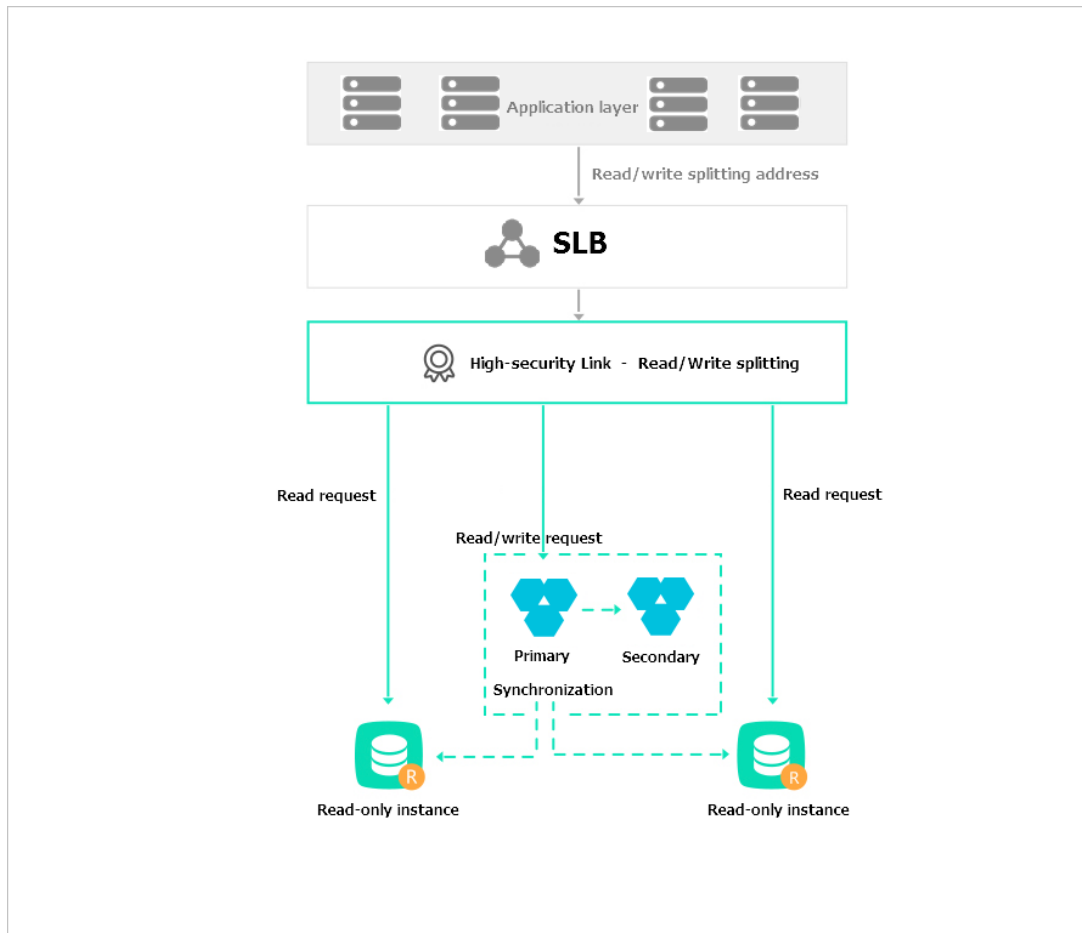
OSS is a secure, reliable, low-cost, and high-capacity storage service offered by Alibaba Cloud. ApsaraDB RDS can be used in conjunction with OSS to implement a multi-type data storage solution. For example, assume that ApsaraDB RDS and OSS are jointly used to set up an online forum. Resources such as the posts and images uploaded to the forum can be stored in OSS to reduce storage needs on ApsaraDB RDS.

8.5.2. Read/write splitting

This feature allows you to split read and write requests across different instances to expand the processing capability of the system.

ApsaraDB RDS for MySQL allows you to directly attach read-only instances to ApsaraDB RDS to reduce read pressure on the primary instance. The primary and read-only instances of ApsaraDB RDS for MySQL each have their own endpoints. After you enable read/write splitting, the system offers a read/write splitting endpoint. This endpoint associates the primary instance with all of its read-only instances to implement automatic read/write splitting, which allows applications to send all read and write requests to a single endpoint. Write requests are routed to the primary instance, and read requests are routed to each read-only instance based on their weights. You can scale up the processing capability of the system by adding more read-only instances, without the need to modify applications. The **Read/write splitting** figure shows the read/write splitting process.

Read/write splitting

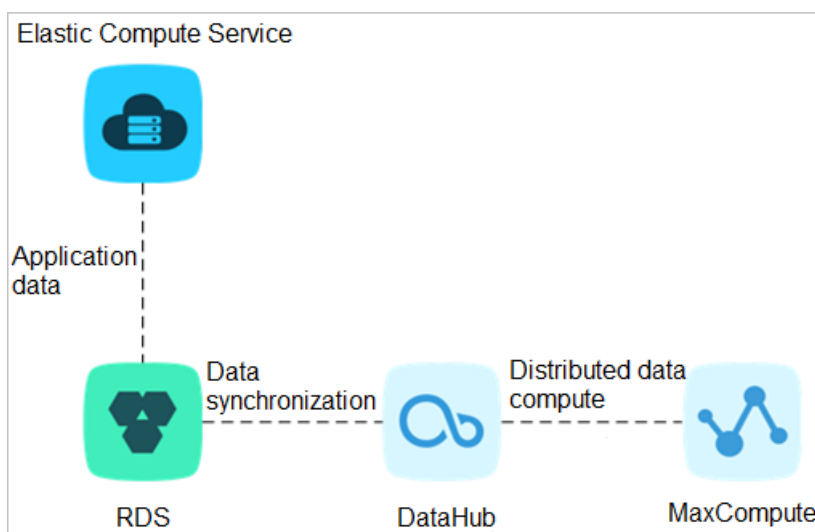


8.5.3. Big data analysis

You can import data from ApsaraDB RDS to MaxCompute to enable large-scale data computing.

MaxCompute is used to store and compute batches of structured data. It provides various data warehouse solutions as well as big data analysis and modeling services, as shown in the **Big data analysis diagram** figure.

Big data analysis diagram



8.6. Limits

8.6.1. Limits on ApsaraDB RDS for MySQL

Before you use ApsaraDB RDS for MySQL, you must understand its limits and take the necessary precautions.

To ensure instance stability and security, ApsaraDB RDS for MySQL has some limits. The [Limits on ApsaraDB RDS for MySQL](#) table describes the limits on ApsaraDB RDS for MySQL.

Limits on ApsaraDB RDS for MySQL

Operation	Limit
Database parameter modification	Most database parameters must be modified by using API operations. For security and stability considerations, only specific parameters can be modified.
Root permissions of databases	The root or system administrator permissions are not provided.
Database backup	- Logical backup can be performed by using the command line interface (CLI) or graphical user interface (GUI). - Physical backup can be performed only by using the ApsaraDB RDS console or API operations.
Database restoration	- Logical restoration can be performed by using the CLI or GUI. - Physical restoration can be performed only by using the ApsaraDB RDS console or API operations.
Database import	- Logical import can be performed by using the CLI or GUI. - Data can be imported by using the MySQL CLI or DTS.
ApsaraDB RDS for MySQL storage engine	- Only InnoDB and TokuDB are supported. Due to the inherent shortcomings of the MyISAM engine, some data may be lost. Only some existing instances use the MyISAM engine. MyISAM engine tables in new instances are converted to InnoDB engine tables. - For performance and security considerations, we recommend that you use the InnoDB storage engine. - The Memory engine is not supported. New Memory tables are converted to InnoDB tables.
Database replication	ApsaraDB RDS for MySQL provides dual-node clusters based on a primary/secondary replication architecture. The secondary instances in this replication architecture are hidden and cannot be directly accessed.
Instance restart	Instances must be restarted by using the ApsaraDB RDS console or API operations.

Operation	Limit
Account and database management	ApsaraDB RDS for MySQL uses the ApsaraDB RDS console to manage accounts and databases. ApsaraDB RDS for MySQL also allows you to create a privileged account to manage users, passwords, and databases.
Standard account	- Custom authorization is not supported. - The ApsaraDB RDS console allows you to manage accounts and databases. - Instances that support standard accounts also support privileged accounts.
Privileged account	- Custom authorization is supported. - The ApsaraDB RDS console does not provide interfaces to manage accounts or databases. These operations can be performed only by using code or DMS. - The privileged account cannot be reverted back to a standard account.

8.6.2. Limits on ApsaraDB RDS for SQL Server

Before you use ApsaraDB RDS for SQL Server, you must understand its limits and take the necessary precautions.

To ensure instance stability and security, ApsaraDB RDS for SQL Server has some limits. The following table describes the limits on ApsaraDB RDS for SQL Server.

Limits on ApsaraDB RDS for SQL Server

Operation	Limit
Maximum number of databases	50
Maximum number of database accounts	500
Creation of users, login accounts, and databases	Supported
Database-level data definition language (DDL) trigger	Limited
Database permission authorization	Limited
Permissions to delete threads	Supported
Linked server	Limited
Distributed transactions	Limited
SQL Profiler	Limited
Tuning Advisor	Limited
Change Data Capture (CDC)	Limited
Change tracking	Supported

Operation	Limit
Windows domain account logon	Limited
Email	Limited
SQL Server Integration Services (SSIS)	Limited
SQL Server Analysis Services (SSAS)	Limited
SQL Server Reporting Services (SSRS)	Limited
R Services	Limited
Common Language Runtime (CLR)	Limited
Asynchronous communication	Limited
Replication	Limited
Policy management	Limited

8.6.3. Limits on ApsaraDB RDS for PostgreSQL

Before you use ApsaraDB RDS for PostgreSQL, you must understand its limits and take the necessary precautions.

The following table describes the limits on ApsaraDB RDS for PostgreSQL.

Operation	Limit
Root permissions of databases	Superuser permissions are not provided.
Database replication	ApsaraDB RDS for PostgreSQL provides a primary/secondary replication architecture except in the Basic Edition. The secondary instances in the architecture are hidden and cannot be accessed by your applications.
Instance restart	Instances must be restarted by using the ApsaraDB RDS console or API operations.

8.6.4. Limits on PolarDB

Before you use PolarDB, you must understand its limits and take the necessary precautions.

The following table describes the limits on PolarDB.

Operation	Limit
Database parameter modification	Not supported.
Root permissions of databases	Superuser permissions are not provided.

Operation	Limit
Database replication	- The system automatically builds HA databases based on PolarDB streaming replication without user input. - Secondary PolarDB instances are hidden and cannot be accessed.
Instance restart	Instances must be restarted by using the ApsaraDB RDS console or API operations.

8.7. Terms

Term	Description
region	The geographical location where the server of your ApsaraDB RDS instance resides. You must specify a region when you create an ApsaraDB RDS instance. The region of an instance cannot be changed after the instance is created. ApsaraDB RDS must be used in conjunction with Elastic Compute Service (ECS) and supports only internal access. Therefore, ApsaraDB RDS instances must be located in the same region as their corresponding ECS instances.
zone	The physical area that has an independent power supply and network in a region. Zones in a region can communicate over internal networks. Network latency for resources within the same zone is lower than that for resources across zones. Faults are isolated between zones. Single-zone deployment refers to the case where three nodes of an ApsaraDB RDS instance are all located in the same zone. Network latency is reduced if an ECS instance and its corresponding ApsaraDB RDS instance are both deployed in the same zone.
instance	The most basic unit of ApsaraDB RDS. An instance is the operating environment of ApsaraDB RDS and works as an independent process on a host. You can create, modify, or delete an ApsaraDB RDS instance in the ApsaraDB RDS console. Instances are independent, and their resources are isolated. They do not compete for resources such as CPU, memory, or I/O. Each instance has its own features, such as database engine and version. ApsaraDB RDS controls instance behavior by using corresponding parameters.
memory	The maximum amount of memory that can be used by an ApsaraDB RDS instance.
disk capacity	The amount of disk space that is selected when you create an ApsaraDB RDS instance. Disk capacity is occupied by the aggregated data and the data required for normal instance operations such as system databases, database rollback logs, redo logs, and indexes. Make sure that the disk capacity is sufficient for the ApsaraDB RDS instance to store data. Otherwise, the ApsaraDB RDS instance may be locked. If the instance is locked due to insufficient disk capacity, you can unlock the instance by expanding the disk capacity.
IOPS	The maximum number of read/write operations performed per second on block devices at a granularity of 4 KB.

Term	Description
CPU core	The maximum computing capability of an ApsaraDB RDS instance. A single Intel Xeon series CPU core has at least 2.3 GHz of computing power with hyper-threading capabilities.
number of connections	The number of TCP connections between a client and an ApsaraDB RDS instance. If the client uses a connection pool, the connection between the client and the ApsaraDB RDS instance is a persistent connection. Otherwise, it is a short-lived connection.

8.8. Instance types

Instances of different editions, versions, and instance types each perform differently from one another.

IOPS

The maximum input/output operations per second (IOPS) of an ApsaraDB RDS instance that uses local SSDs varies based on the instance type. The maximum IOPS of an ApsaraDB RDS instance that uses standard SSDs or enhanced SSDs (ESSDs) varies based on the instance type and storage capacity. The following formula can be used to calculate the IOPS of an ApsaraDB RDS instance that uses standard SSDs or ESSDs. The storage capacity is measured in GB.

$$\min\{1800 + 30 \times \text{Storage capacity}, 25000\}$$

 **Note** If the throughput of an instance reaches the upper limit, the IOPS of the instance also decreases.

ApsaraDB RDS for MySQL (High-availability Edition with local SSDs)

RDS edition	Instance family	Instance type	CPU and memory	Maximum connections	Storage	
					Maximum IOPS	Storage capacity
		5.6, 5.7, and 8.0 rds.mysql.t1.small	1 core, 1 GB	300	600	
		5.6, 5.7, and 8.0 rds.mysql.s1.small	1 core, 2 GB	600	1,000	
		5.6, 5.7, and 8.0 rds.mysql.s2.large	2 cores, 4 GB	1,200	2,000	
		5.6, 5.7, and 8.0 rds.mysql.s2.xlarge	2 cores, 8 GB	2,000	4,000	

RDS edition	Instance family	Instance type	CPU and memory	Maximum connections	Storage	5–2,000 GB
					Maximum IOPS	Storage capacity
High-availability Edition	General-purpose	5.6, 5.7, and 8.0 rds.mysql.s3.large	4 cores, 8 GB	2,000	5,000	
		5.6, 5.7, and 8.0 rds.mysql.m1.medium	4 cores, 16 GB	4,000	7,000	
		5.6, 5.7, and 8.0 rds.mysql.c1.large	8 cores, 16 GB	4,000	8,000	
		5.6, 5.7, and 8.0 rds.mysql.c1.xlarge	8 cores, 32 GB	8,000	12,000	
		5.6, 5.7, and 8.0 rds.mysql.c2.xlarge	16 cores, 64 GB	16,000	14,000	5–3,000 GB
		5.6, 5.7, and 8.0 rds.mysql.c2.xlp2	16 cores, 96 GB	24,000	16,000	
		5.6 and 5.7 rds.mysql.c2.2xlarge	16 cores, 128 GB	32,000	16,000	
		5.6, 5.7, and 8.0 mysql.x4.large.2	4 cores, 16 GB	2,500	4,500	50–1,000 GB
		5.6, 5.7, and 8.0 mysql.x4.xlarge.2	8 cores, 32 GB	5,000	9,000	500–3,000 GB
		5.6, 5.7, and 8.0 mysql.x4.2xlarge.2	16 cores, 64 GB	10,000	18,000	
		5.6, 5.7, and 8.0 mysql.x4.4xlarge.2	32 cores, 128 GB	20,000	36,000	
		5.6, 5.7, and 8.0 mysql.x8.medium.2	2 cores, 16 GB	2,500	4,500	

RDS edition	Instance family	Instance type	CPU and memory	Maximum connections	Storage	
					Maximum IOPS	Storage capacity
		5.6, 5.7, and 8.0 mysql.x8.large.2	4 cores, 32 GB	5,000	9,000	50-1,000 GB
		5.6, 5.7, and 8.0 mysql.x8.xlarge.2	8 cores, 64 GB	10,000	18,000	
		5.6, 5.7, and 8.0 mysql.x8.2xlarge.2	16 cores, 128 GB	20,000	36,000	
		5.6, 5.7, and 8.0 mysql.x8.4xlarge.2	32 cores, 256 GB	40,000	72,000	500-3,000 GB
		5.6, 5.7, and 8.0 mysql.x8.8xlarge.2	64 cores, 512 GB	80,000	144,000	
	Dedicated host	5.6, 5.7, and 8.0 rds.mysql.st.v52	90 cores, 720 GB	150,000	140,000	1,000-6,000 GB
		5.6 and 5.7 rds.mysql.st.h43	60 cores, 470 GB	150,000	120,000	

ApsaraDB RDS for MySQL (Enterprise Edition with local SSDs)

RDS edition	Instance family	Instance type	CPU and memory	Maximum connections	Storage	
					Maximum IOPS	Storage capacity
		5.7 mysql.n2.small.25	90 cores, 720 GB	150,000	1,000	1,000-6,000 GB
		5.7 mysql.n2.medium.25	2 cores, 4 GB	1,200	2,000	
		5.7 mysql.n4.medium.25	2 cores, 8 GB	2,000	4,000	

RDS edition	Instance family	Instance type	CPU and memory	Maximum connections	Storage	Storage capacity 5–2,000 GB
					Maximum IOPS	
Enterprise Edition	General-purpose	5.7 mysql.n2.large.25	4 cores, 8 GB	2,000	5,000	
		5.7 mysql.n4.large.25	4 cores, 16 GB	4,000	7,000	
		5.7 mysql.n2.xlarge.25	8 cores, 16 GB	4,000	8,000	
		5.7 mysql.n4.xlarge.25	8 cores, 32 GB	8,000	12,000	
		5.7 mysql.n4.2xlarge.25	16 cores, 64 GB	16,000	14,000	5–3,000 GB
		5.7 mysql.n8.2xlarge.25	16 cores, 128 GB	32,000	16,000	
	Dedicated (with a large number of cores)	5.7 mysql.x4.large.25	4 cores, 16 GB	2,500	4,500	50–1,000 GB
		5.7 mysql.x4.xlarge.25	8 cores, 32 GB	5,000	9,000	500–3,000 GB
		5.7 mysql.x4.2xlarge.25	16 cores, 64 GB	10,000	18,000	
		5.7 mysql.x4.4xlarge.25	32 cores, 128 GB	20,000	36,000	1,000–3,000 GB
		5.7 mysql.x8.medium.25	2 cores, 16 GB	2,500	4,500	50–1,000 GB
		5.7 mysql.x8.large.25	4 cores, 32 GB	5,000	9,000	

RDS edition	Instance family (with memory capacity)	Instance type	CPU and memory	Maximum connections	Storage	Storage capacity
					Maximum IOPS	
		5.7 mysql.x8.xlarge.25	8 cores, 64 GB	10,000	18,000	500-3,000 GB
		5.7 mysql.x8.2xlarge.25	16 cores, 128 GB	20,000	36,000	
		5.7 mysql.x8.4xlarge.25	32 cores, 256 GB	40,000	72,000	1,000-3,000 GB
	Dedicated host	5.7 mysql.st.8xlarge.25	60 cores, 470 GB	100,000	120,000	1,000-6,000 GB
		5.7 mysql.st.12xlarge.25	90 cores, 720 GB	150,000	140,000	

Read-only ApsaraDB RDS for MySQL instances (with local SSDs)

Instance role	Instance family	Instance type	CPU and memory	Maximum connections	Storage	
					Maximum IOPS	Storage capacity
		5.6, 5.7, and 8.0 rds.mysql.t1.small	1 core, 1 GB	300	600	5-2,000 GB
		5.6, 5.7, and 8.0 rds.mysql.s1.small	1 core, 2 GB	600	1,000	
		5.6, 5.7, and 8.0 rds.mysql.s2.large	2 cores, 4 GB	1,200	2,000	
		5.6, 5.7, and 8.0 rds.mysql.s2.xlarge	2 cores, 8 GB	2,000	4,000	
		5.6, 5.7, and 8.0 rds.mysql.s3.large	4 cores, 8 GB	2,000	5,000	

Instance role	General instance family	Instance type	CPU and memory	Maximum connections	Storage	Storage capacity
					Maximum IOPS	
Read-only instance		5.6, 5.7, and 8.0 rds.mysql.m1.medium	4 cores, 16 GB	4,000	7,000	
		5.6, 5.7, and 8.0 rds.mysql.c1.large	8 cores, 16 GB	4,000	8,000	
		5.6, 5.7, and 8.0 rds.mysql.c1.xlarge	8 cores, 32 GB	8,000	12,000	
		5.6, 5.7, and 8.0 rds.mysql.c2.xlarge	16 cores, 64 GB	16,000	14,000	5-3,000 GB
		5.6, 5.7, and 8.0 rds.mysql.c2.xlp2	16 cores, 96 GB	24,000	16,000	
	Dedicated	5.6, 5.7, and 8.0 mysqlro.x4.large.1	4 cores, 16 GB	2,500	4,500	50-1,000 GB
		5.6, 5.7, and 8.0 mysqlro.x4.xlarge.1	8 cores, 32 GB	5,000	9,000	
		5.6, 5.7, and 8.0 mysqlro.x4.2xlarge.1	16 cores, 64 GB	10,000	18,000	500-3,000 GB
		5.6, 5.7, and 8.0 mysqlro.x4.4xlarge.1	32 cores, 128 GB	20,000	36,000	
		5.6, 5.7, and 8.0 mysqlro.x8.medium.1	2 cores, 16 GB	2,500	4,500	50-1,000 GB
		5.6, 5.7, and 8.0 mysqlro.x8.large.1	4 cores, 32 GB	5,000	9,000	
		5.6, 5.7, and 8.0 mysqlro.x8.xlarge.1	8 cores, 64 GB	10,000	18,000	

Instance role	Instance family	Instance type	CPU and memory	Maximum connections	Storage	500–3,000 GB Storage capacity
					Maximum IOPS	
		5.6, 5.7, and 8.0 mysqlro.x8.2xlarge.1	16 cores, 128 GB	20,000	36,000	1,000–6,000 GB
		5.6, 5.7, and 8.0 mysqlro.x8.4xlarge.1	32 cores, 256 GB	40,000	72,000	
		5.6, 5.7, and 8.0 mysqlro.x8.8xlarge.1	64 cores, 512 GB	80,000	144,000	
	Dedicated host	5.6, 5.7, and 8.0 rds.mysql.st.v52	90 cores, 720 GB	150,000	140,000	1,000–6,000 GB
		5.6 and 5.7 rds.mysql.st.h43	60 cores, 470 GB	100,000	120,000	

ApsaraDB RDS for PostgreSQL (High-availability Edition with local SSDs)

RDS edition	Instance family	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Storage capacity
	Dedicated (with a large memory capacity)	9.4 and 10.0 pg.x8.medium.2	2 cores, 16 GB	2,500	4,500	
		9.4 and 10.0 pg.x8.large.2	4 cores, 32 GB	5,000	9,000	
		9.4 and 10.0 pg.x8.xlarge.2	8 cores, 64 GB	10,000	18,000	
		9.4 and 10.0 pg.x8.2xlarge.2	16 cores, 128 GB	12,000	36,000	

• High-availability Edition	Instance family	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Storage capacity (in 5 GB increments)
• Read-only instance	Dedicated (with a large number of cores)	9.4 and 10.0 pg.x4.large.2	4 cores, 16 GB	2,500	4,500	20-6,900 GB
		9.4 and 10.0 pg.x4.xlarge.2	8 cores, 32 GB	5,000	9,000	
		9.4 and 10.0 pg.x4.2xlarge.2	16 cores, 64 GB	10,000	18,000	
		9.4 and 10.0 pg.x4.4xlarge.2	32 cores, 128 GB	12,000	36,000	
	Dedicated host	9.4 and 10.0 rds.pg.st.h4.3	60 cores, 470 GB	4,000	50,000	

ApsaraDB RDS for PostgreSQL (High-availability Edition with standard SSDs or ESSDs)

RDS edition	Instance family	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Storage capacity
	General-purpose	10.0, 11.0, and 12.0 pg.n2.medium.2c	2 cores, 4 GB	100		
		10.0, 11.0, and 12.0 pg.x4.medium.2c	2 cores, 8 GB	800		
		10.0, 11.0, and 12.0 pg.x8.medium.2c	2 cores, 16 GB	1,600		
		10.0, 11.0, and 12.0 pg.x2.large.2c	4 cores, 8 GB	800		

RDS edition	Instance family	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Storage capacity
High-availability Edition	Dedicated	10.0, 11.0, and 12.0 pg.x4.large.2c	4 cores, 16 GB	1,600	IOPS	20-6,000 GB
		10.0, 11.0, and 12.0 pg.x8.large.2c	4 cores, 32 GB	3,200		
		10.0, 11.0, and 12.0 pg.x2.xlarge.2c	8 cores, 16 GB	1,600		
		10.0, 11.0, and 12.0 pg.x4.xlarge.2c	8 cores, 32 GB	3,200		
		10.0, 11.0, and 12.0 pg.x8.xlarge.2c	8 cores, 64 GB	6,400		
		10.0, 11.0, and 12.0 pg.x2.3large.2c	12 cores, 24 GB	2,400		
		10.0, 11.0, and 12.0 pg.x4.3large.2c	12 cores, 48 GB	4,800		
		10.0, 11.0, and 12.0 pg.x8.3large.2c	12 cores, 96 GB	9,600		
		10.0, 11.0, and 12.0 pg.x2.2xlarge.2c	16 cores, 32 GB	3,200		
		10.0, 11.0, and 12.0 pg.x4.2xlarge.2c	16 cores, 64 GB	6,400		
		10.0, 11.0, and 12.0 pg.x8.2xlarge.2c	16 cores, 128 GB	12,800		
		10.0, 11.0, and 12.0 pg.x2.3xlarge.2c	24 cores, 48 GB	4,800		

RDS edition	Instance family	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Storage capacity
		10.0, 11.0, and 12.0 pg.x4.3xlarge.2c	24 cores, 96 GB	9,600		
		10.0, 11.0, and 12.0 pg.x8.3xlarge.2c	24 cores, 192 GB	19,200		
		10.0, 11.0, and 12.0 pg.x2.4xlarge.2c	32 cores, 64 GB	6,400		
		10.0, 11.0, and 12.0 pg.x4.4xlarge.2c	32 cores, 128 GB	12,800		
		10.0, 11.0, and 12.0 pg.x8.4xlarge.2c	32 cores, 256 GB	25,600		
		10.0, 11.0, and 12.0 pg.x2.13large.2c	52 cores, 104 GB	10,400		
		10.0, 11.0, and 12.0 pg.x4.13large.2c	52 cores, 192 GB	19,200		
		10.0, 11.0, and 12.0 pg.x8.13large.2c	52 cores, 384 GB	38,400		
		10.0, 11.0, and 12.0 pg.x2.8xlarge.2c	64 cores, 128 GB	12,800		
		10.0, 11.0, and 12.0 pg.x4.8xlarge.2c	64 cores, 256 GB	25,600		
		10.0, 11.0, and 12.0 pg.x8.8xlarge.2c	64 cores, 512 GB	51,200		
		10.0, 11.0, and 12.0 pg.x2.13xlarge.2c	104 cores, 192 GB	19,200		

RDS edition	Instance family	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Storage capacity
		10.0, 11.0, and 12.0 pg.x4.13xlarge.2c	104 cores, 384 GB	38,400		
		10.0, 11.0, and 12.0 pg.x8.13xlarge.2c	104 cores, 768 GB	76,800		

ApsaraDB RDS for SQL Server (High-availability Edition with local SSDs)

RDS edition	SQL Server version	Instance family	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Storage capacity
	2008 R2	Dedicated	2008r2 mssql.x8.medium.2	2 cores, 16 GB	2,500	4,500	250 GB
			2008r2 mssql.x8.large.2	4 cores, 32 GB	5,000	9,000	500 GB
			2008r2 mssql.x8.xlarge.2	8 cores, 64 GB	10,000	18,000	1,000 GB
			mssql.x8.2xlarge.2	16 cores, 128 GB	20,000	36,000	2,000 GB
		Dedicated host	2008r2 rds.mssql.st.d13	30 cores, 220 GB	64,000	20,000	2 TB
			2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x4.medium.e2	2 cores, 8 GB			
			2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x8.medium.e2	2 cores, 16 GB			

RDS edition	SQL Server version	Instance family	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Storage capacity
High-availability Edition	2012 EE: 2012_ent_ha 2016 EE: 2016_ent_ha 2017 EE	Dedicated	2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x4.large.e2	4 cores, 16 GB			
			2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x8.large.e2	4 cores, 32 GB			
			2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x4.xlarge.e2	8 cores, 32 GB			
			2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x8.xlarge.e2	8 cores, 64 GB			
			2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x4.2xlarge.e2	16 cores, 64 GB			
			2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x8.2xlarge.e2	16 cores, 128 GB			
			2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x4.3xlarge.e2	24 cores, 96 GB			
			2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x4.4xlarge.e2	32 cores, 128 GB			

RDS edition	SQL Server version	Instance family	Instance type	CPU and memory	Maximum connections	For more information, see the "IOPS" section.	Storage capacity
			2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x8.4xlarge.e2	32 cores, 256 GB	Unlimited		20–4,000 GB
			2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x8.7xlarge.e2	56 cores, 480 GB			
			2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x4.8xlarge.e2	64 cores, 256 GB			
			2012_ent_ha, 2016_ent_ha, and 2017_ent mssql.x8.8xlarge.e2	64 cores, 512 GB			
	2012 SE: 2012_std_ha	Dedicated	2012_std_ha and 2016_std_ha mssql.x4.medium.s2	2 cores, 8 GB			
			2012_std_ha and 2016_std_ha mssql.x8.medium.s2	2 cores, 16 GB			
			2012_std_ha and 2016_std_ha mssql.x4.large.s2	4 cores, 16 GB			
			2012_std_ha and 2016_std_ha mssql.x8.large.s2	4 cores, 32 GB			
			2012_std_ha and 2016_std_ha mssql.x4.xlarge.s2	8 cores, 32 GB			

RDS edition	2016 SQL Server 2016 version std_ha	Instance family	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Storage capacity
			2012_std_ha and 2016_std_ha mssql.x8.xlarge.s2	8 cores, 64 GB			
			2012_std_ha and 2016_std_ha mssql.x4.2xlarge.s2	16 cores, 64 GB			
			2012_std_ha and 2016_std_ha mssql.x8.2xlarge.s2	16 cores, 128 GB			
			2012_std_ha and 2016_std_ha mssql.x4.3xlarge.s2	24 cores, 96 GB			

PolarDB (High-availability Edition)

RDS edition	Instance family	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Storage capacity
		11 polardb.x4.small.2	1 core, 4 GB	200	5,000	250 GB
		11 polardb.x4.medium.2	2 cores, 8 GB	400	10,000	
		11 polardb.x8.medium.2	2 cores, 16 GB	2,500	15,000	
		11 polardb.x4.large.2	4 cores, 16 GB	2,500	20,000	250-500 GB (in 5 GB increments)
		11 polardb.x8.large.2	4 cores, 32 GB	5,000	30,000	

RDS edition	Instance family	Instance type	CPU and memory	Maximum connections	Maximum IOPS	Storage capacity
- High-availability Edition - Read-only instance	Dedicated	11 polardb.x4.xlarge.2	8 cores, 32 GB	5,000	40,000	500–1,000 GB (in 5 GB increments)
		11 polardb.x8.xlarge.2	8 cores, 64 GB	10,000	60,000	
		11 polardb.x4.2xlarge.2	16 cores, 64 GB	10,000	80,000	1,000–2,000 GB (in 5 GB increments)
		11 polardb.x8.2xlarge.2	16 cores, 128 GB	12,000	120,000	
		11 polardb.x4.4xlarge.2	32 cores, 128 GB	12,000	160,000	2,000–3,000 GB (in 5 GB increments)
		11 polardb.x8.4xlarge.2	32 cores, 256 GB	12,000	240,000	
	Dedicated host	11 polardb.x8.12xlarge.2	88 cores, 710 GB	12,000	450,000	3,000–6,000 GB
		11 rds.polardb.st.h43	60 cores, 470 GB	12,000	450,000	6,000 GB

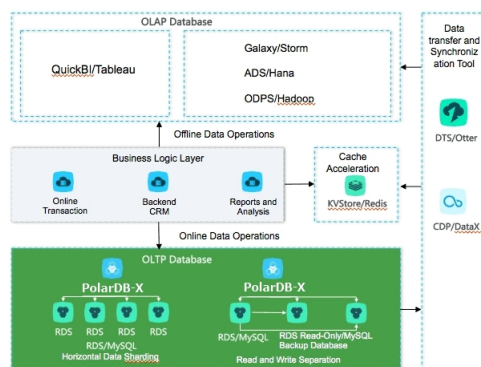
9.Cloud Native Distributed Database PolarDB-X

9.1. What is PolarDB-X?

Cloud Native Distributed Database PolarDB-X (PolarDB-X) is a middleware service independently developed by Alibaba Group for scale-out of single-instance relational databases. It is compatible with Distributed Relational Database Service (DRDS). Compatible with the MySQL protocol, PolarDB-X supports most MySQL data manipulation language (DML) and data definition language (DDL) syntax. It provides the core capabilities of distributed databases, such as database sharding, table sharding, smooth scale-out, configuration changing, and transparent read/write splitting. PolarDB-X features lightweight (stateless), flexibility, stability, and efficiency, and provides you with O&M capabilities throughout the lifecycle of distributed databases.

PolarDB-X is mainly used for operations on large-scale online data. By splitting data in specific business scenarios, PolarDB-X maximizes the operation efficiency, meeting the requirements of online businesses on relational databases.

Product structure diagram of PolarDB-X



Problems solved

- Capacity bottleneck of single-instance databases: As the data volume and access volume increase, traditional single-instance databases encounter great challenges that cannot be completely solved by hardware upgrades. Distributed solutions use multiple instances to work jointly, effectively resolving the bottlenecks of data storage capacity and access volumes.
- Difficult scale-out of relational databases: Due to the inherent attributes of distributed databases, data can be stored to different shards through smooth data migration, supporting the dynamic scale-out of relational databases.

9.2. Benefits

This topic describes the benefits of PolarDB-X.

Distributed architecture

PolarDB-X enables horizontal splitting of data and the cluster deployment of a single service. This resolves single-instance bottlenecks of Server Load Balancer (SLB), Cloud Native Distributed Database PolarDB-X, and ApsaraDB RDS for MySQL. In this case, service scalability is achieved. This is also one of the key benefits of distributed databases.

Audit logs

Audit logs are stored in Log Service (SLS) in an automatic way. You can download logs from SLS. This helps you store logs for a long time and manage audit logs.

Auto scaling

PolarDB-X instances and ApsaraDB RDS for MySQL instances can be added or removed in a dynamic way. This enables horizontal scaling. PolarDB-X is scalable and compatible with the MySQL ecosystem. You can scale out cluster nodes in a linear way. The overall performance increases based on the number of nodes. In this case, you can select service capabilities in a flexible way.

High performance

PolarDB-X that is integrated with ApsaraDB RDS for MySQL splits data in specific business scenarios and enables data clustering based on major operations. Therefore, it speeds up the response to online transaction operations. PolarDB-X

High-availability clusters

PolarDB-X adopts a multi-node cluster architecture. Each component management node on the platform implements a high-availability mechanism. Therefore, the failure of a management node for operations and maintenance (O&M) in a cluster does not affect the service availability. If a service node in a cluster fails, the overall instance performance is not affected. PolarDB-X also provides the adaptive load balancing capability for clusters. To ensure service availability in high-concurrency and heavy load request scenarios, you do not need to change cluster parameters.

Backup

PolarDB-X stores data in ApsaraDB RDS for MySQL instances that support full or incremental data backup and data recovery from storage. PolarDB-X supports cluster data backup between data centers. This meets the mutual data backup requirements between data centers. You can manage data backup processes in a visualized manner.

Disaster recovery

Metadatabases support fast switchover for disaster recovery. Single-node failures do not affect services. This achieves zero downtime for service failover.

Security and controllability

PolarDB-X supports an account and permission system similar to that of single-instance databases. PolarDB-X enables useful features, such as the IP address whitelist and default disabling of high-risk SQL statements. PolarDB-X provides a complete standard API system that can be integrated with your on-premises management system. Complete product support and architecture services are also available to you.

Isolation

By using the multi-instance method, PolarDB-X can process tasks of multiple tenants in parallel. Tenant tasks are submitted to queues on different instances for execution. PolarDB-X isolates tenant resources

by isolating PolarDB-X instances.

Permissions

- PolarDB-X allows you to manage tenants in a centralized manner. You can configure and manage tenant resources in a dynamic manner, isolate resources, view the resource usage, and manage tenants at multiple levels in the console.
- PolarDB-X supports permission management and fine-grained audit of user operations.
- PolarDB-X provides comprehensive authentication and isolation mechanisms to ensure user data privacy.
- PolarDB-X allows you to isolate resources by using the multi-tenant mode.

Scheduling

PolarDB-X supports multi-tenant scheduling in multiple clusters and resource pools.

9.3. Architecture

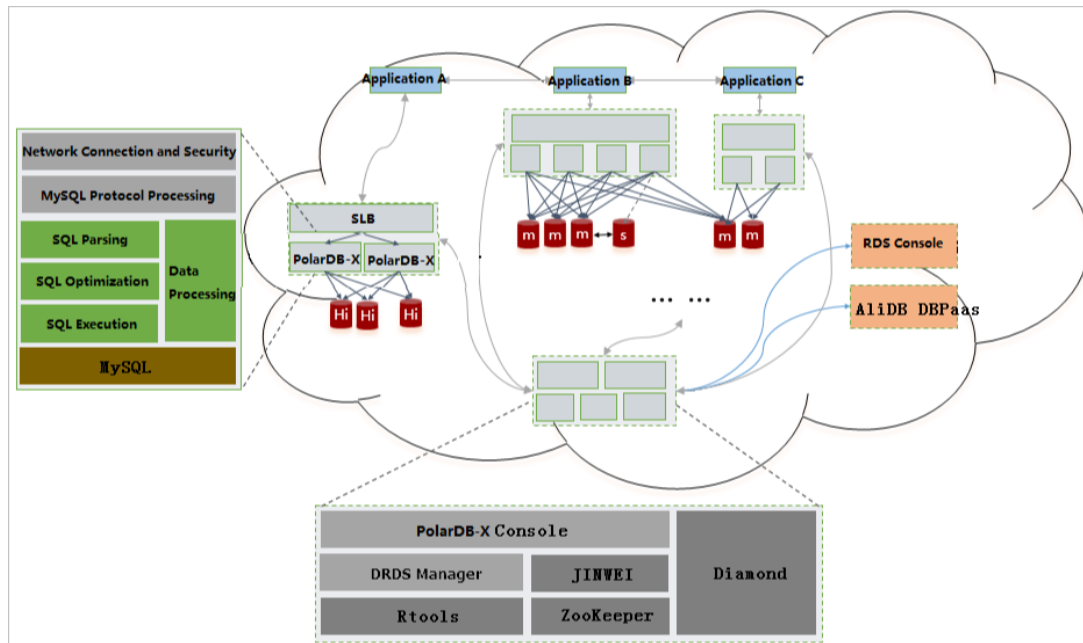
PolarDB-X supports two output methods: integrated output by Apsara Stack and separate output by Alibaba middleware. The two output methods differ in the features and the components that PolarDB-X depends on.

The following table describes the differences between the two methods.

Difference	Integrated output by Apsara Stack	Separate output by Alibaba middleware
MySQL	RDS for MySQL	Alibaba Group Database Platform as a Service (DBPaaS)
Load balancing	Centralized Server Load Balancer (SLB)	Client-side load balancer (VIPServer)
Support for special storage	None	

The following figure shows the system architecture of PolarDB-X.

PolarDB-X system architecture



PolarDB-X Server

PolarDB-X Server is the service layer of PolarDB-X. Multiple service nodes constitute a service cluster to provide distributed database services, including read/write splitting, routed SQL execution, result combination, dynamic database configuration, and globally unique identifiers (GUIDs).

Note PolarDB-X instances are stateless nodes. PolarDB-X uses ApsaraDB RDS for MySQL instances for data storage. PolarDB-X encrypts data by using encryption algorithms, including SHA-256 and AES-128.

ApsaraDB RDS for MySQL (represented by m and s in the figure)

ApsaraDB RDS for MySQL stores data and performs online data operations. It achieves high availability by using MySQL primary/secondary replication. ApsaraDB RDS for MySQL also enables dynamic database failover by using the primary/secondary switchover mechanism.

In the ApsaraDB RDS for MySQL console, you can manage and monitor instances, and manage alerts and resources throughout the instance lifecycle.

HiStore

DBPaaS

If you select the separate output of PolarDB-X instead of the integrated output by Apsara Stack, you can perform operations on the built-in MySQL O&M platform DBPaaS. For example, you can manage and monitor instances, and manage alerts and resources throughout the instance lifecycle.

Centralized SLB

You do not need to install a client on your instance. SLB is used to distribute your requests. When an instance fails or a new instance is added, SLB ensures that traffic on the storage nodes is evenly distributed.

VIPServer that enables client-side load balancing

You must install a client on your instance to reduce dependency on the central controller. This indicates that interaction is performed only when the load configuration changes. VIPServer is used to distribute the traffic of your requests. When an instance fails or a new instance is added, VIPServer ensures that traffic on the storage nodes is evenly distributed.

Diamond

Diamond is a system that allows you to configure storage and manage instances for PolarDB-X. You can configure storage settings, query instance information, and subscribe to notifications. In PolarDB-X, Diamond stores the source data of databases and configurations that include sharding rules and PolarDB-X switches.

Data Replication System

Data Replication System migrates and synchronizes data for PolarDB-X. Its core capabilities include full data migration and incremental data synchronization. Its derived capabilities include smooth data import, smooth scale-out, and global secondary indexes. Data Replication System requires the support of ZooKeeper and PolarDB-X Rtools.

PolarDB-X Console for user operations

PolarDB-X Console is designed for database administrators (DBAs) to isolate resources and operations based on users. It enables multiple features, such as instance management, database and table management, read/write splitting configuration, smooth scale-out, monitoring display, and the IP address whitelist.

PolarDB-X Manager for O&M operations

PolarDB-X Manager is designed for global O&M operations and DBAs. It enables resource management and system monitoring for PolarDB-X. The core features contain the following two aspects:

- Manage all the resources on which ApsaraDB RDS for MySQL instances depend, including virtual machines, load balancers, and domain names.
- Monitor the PolarDB-X instance status, including the queries per second (QPS), active threads, the number of connections, node network I/O, and node CPU utilization.

Rtools

Rtools is the O&M support system of PolarDB-X. It allows you to manage database configurations, read/write weights, connection parameters, topologies of databases and tables, and sharding rules.

9.4. Features

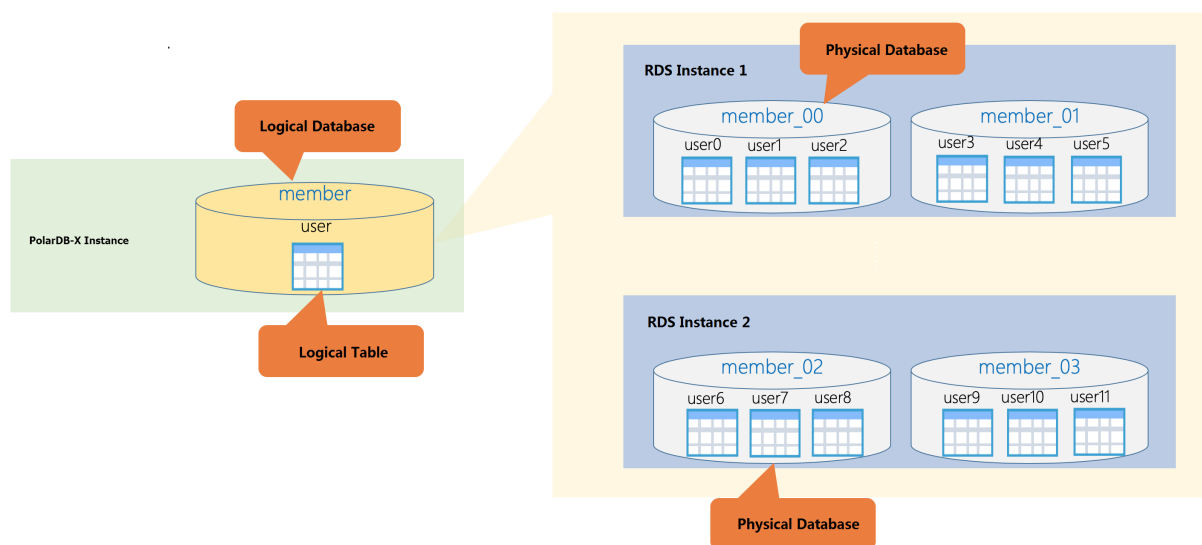
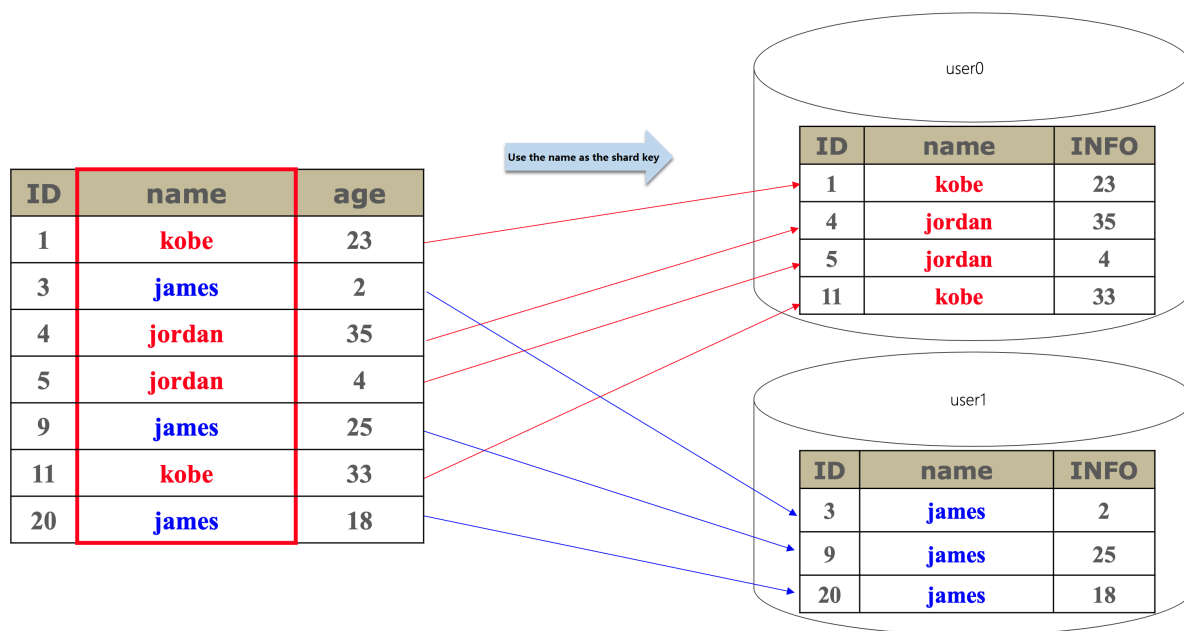
9.4.1. Scalability

Concurrency and storage capacity scalability

The essence of scalability lies in splitting. PolarDB-X distributes data to multiple ApsaraDB RDS for MySQL instances to obtain the distribution of read/write requests and storage through **Horizontal partitioning**. The PolarDB-X layer is stateless and increases nodes to cope with concurrent SQL loads, which is similar to a business application.

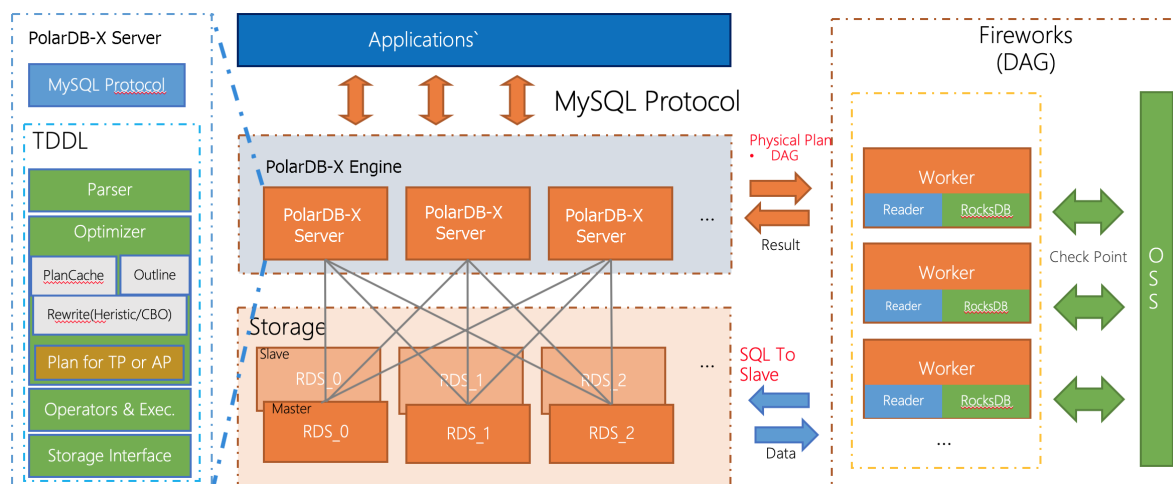
Horizontal partitioning

Data is distributed to multiple ApsaraDB RDS for MySQL instances based on certain calculation and routing rules. In fact, PolarDB-X has many algorithms to cope with the loads in various scenarios.



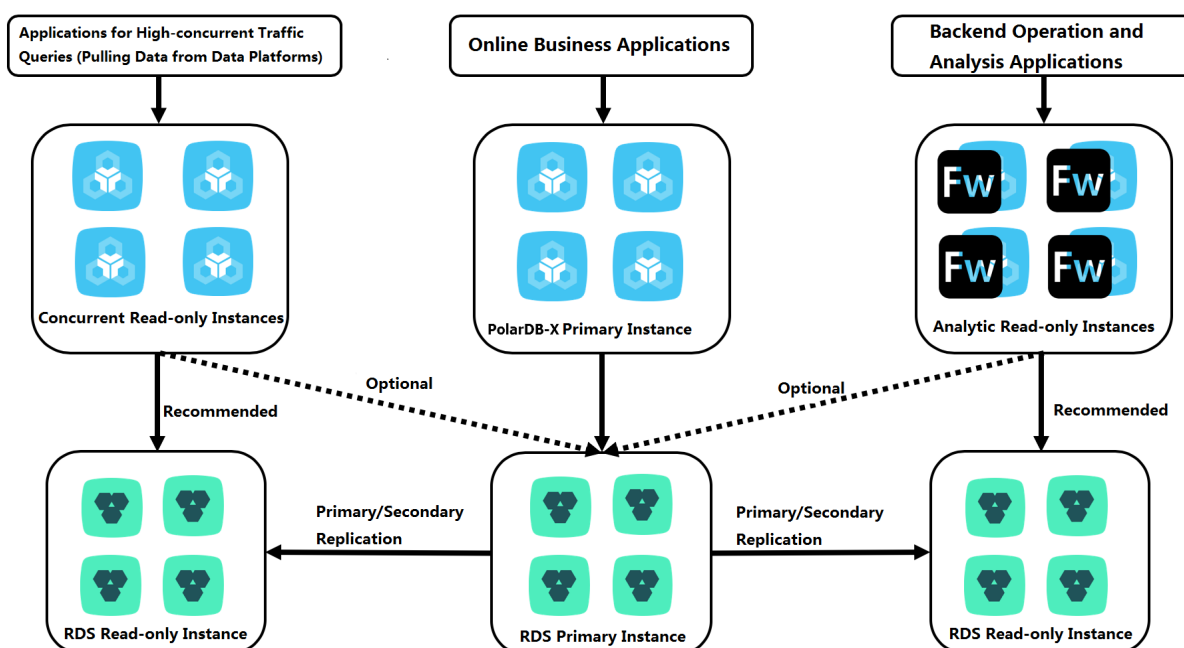
Computing scalability

PolarDB-X often needs to perform complex computing on data far exceeding the capacity of a single instance. These SQL statements include multi-table join, multi-layer nested subqueries, grouping, sorting, and aggregation.



To process complex SQL statements in the online databases, PolarDB-X has expanded the Symmetric Multi-Processing (SMP) and Massively Parallel Processing with Directed Acyclic Graph (MPP&DAG). SMP is fully integrated into the PolarDB-X kernel, while MPP&DAG of PolarDB-X builds a computing cluster that dynamically obtains execution plans for distributed computing at runtime and improves the computing capability by adding nodes.

Currently, the PolarDB-X instances that process data on multiple instances in parallel are provided for businesses in the form of analytic read-only instances.



9.4.2. Distributed transactions

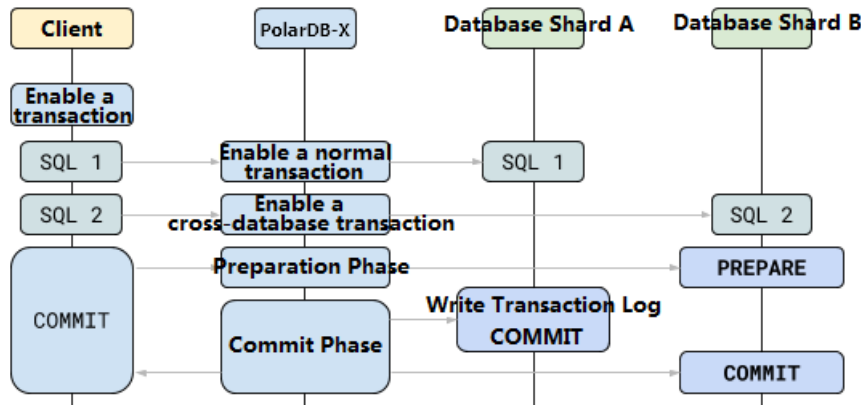
Distributed transactions use Two-Phase Commit (2PC) to ensure the atomicity and consistency of transactions.

A 2PC transaction is divided into the PREPARE phase and the COMMIT phase.

- In the PREPARE phase, data nodes prepare all the resources required for committing transactions, such as locking and logging.

- In the COMMIT phase, data nodes commit transactions.

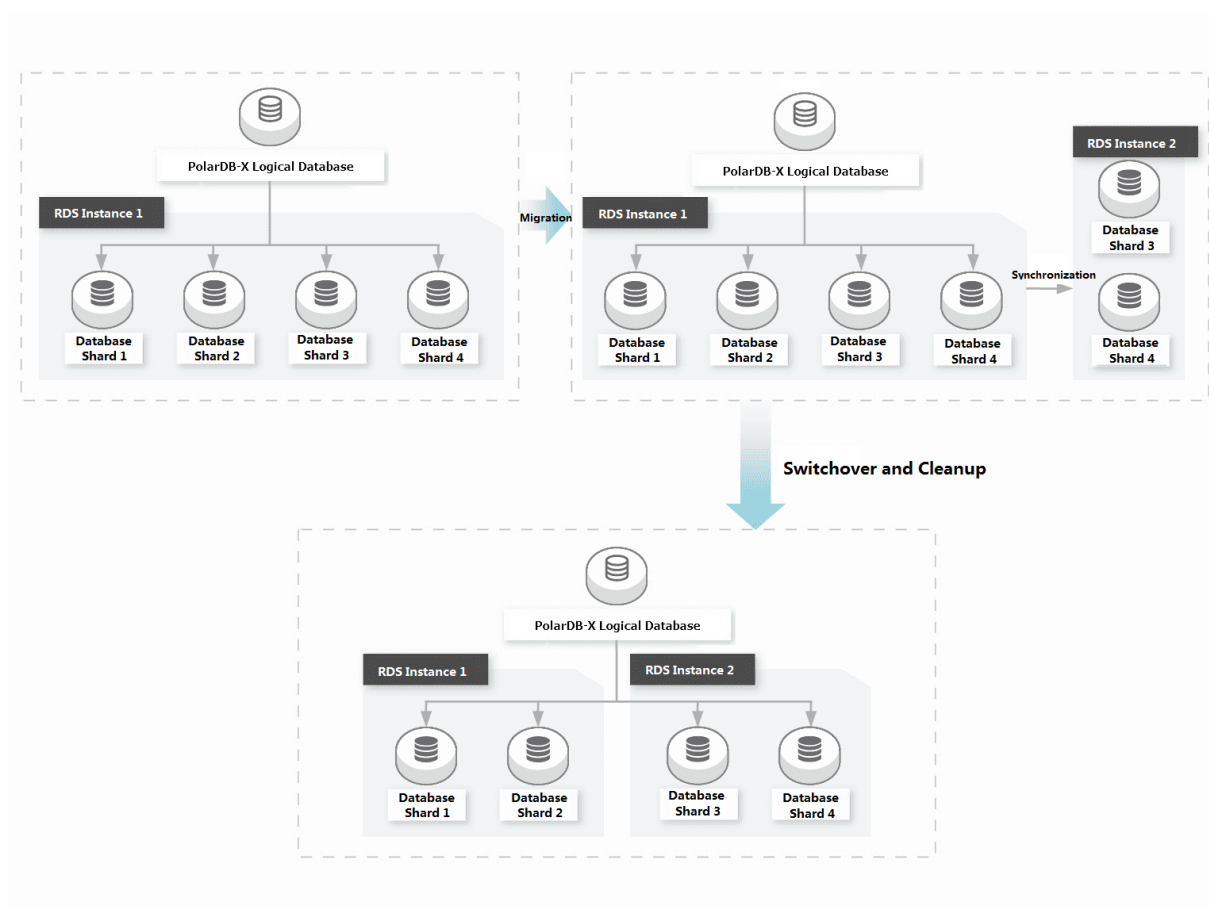
When you commit a distributed transaction, the PolarDB-X server, as a transaction manager, sends a COMMIT request to each data node only after all data nodes (MySQL servers) have their resources ready in PREPARE phase.



9.4.3. Smooth scale-out

When the underlying storage of the logical database reaches the physical bottleneck, for example, when the remaining disk space is about 30%, you can smoothly scale it out to improve the performance.

Smooth scale-out is an online horizontal expansion method. It smoothly migrates the original database shards to the new ApsaraDB RDS for MySQL instances and increases the overall data storage capacity by adding ApsaraDB RDS for MySQL instances, which reduces the pressure on each RDS instance to process data.



9.4.4. Read/write splitting

When a primary ApsaraDB RDS for MySQL instance is heavily loaded with many read requests, you can use the read/write splitting function of PolarDB-X to distribute the read traffic, which reduces the read pressure on the primary ApsaraDB RDS for MySQL instance.

The read/write splitting function of PolarDB-X is transparent to applications. The read traffic can be distributed to the primary ApsaraDB RDS for MySQL instance and multiple ApsaraDB RDS for MySQL read-only instances according to the read weight set in the PolarDB-X console, without changing any code of the application. All the write traffic is distributed to the primary ApsaraDB RDS for MySQL instance.

After read/write splitting is set, real-time strong consistency can be implemented when data is read from the primary ApsaraDB RDS for MySQL instance. Data on the read-only instances is replicated asynchronously from the primary ApsaraDB RDS for MySQL instance, with a millisecond-level latency, therefore real-time strong consistency cannot be implemented when data is read from read-only ApsaraDB RDS for MySQL instances. For SQL statements which require real time and strong consistency for reading data, specify the primary ApsaraDB RDS for MySQL instance to execute these statements through hints of PolarDB-X.

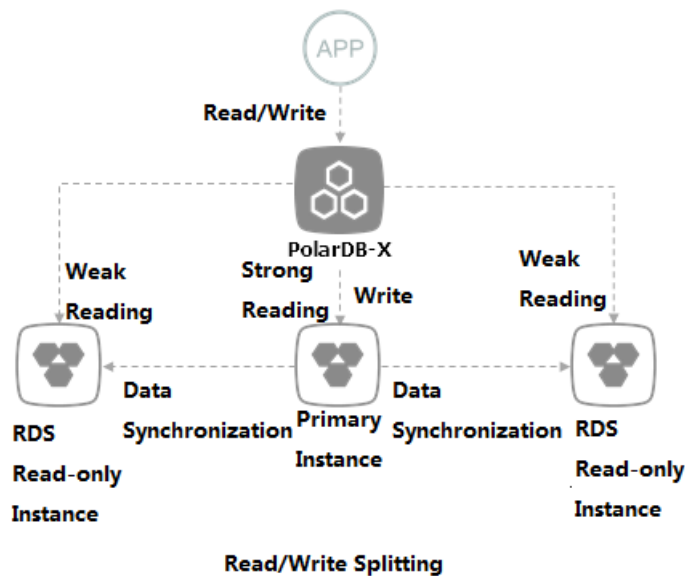
Read/write splitting in non-partition mode

In non-partition mode, PolarDB-X can implement read/write splitting without horizontal partitioning. When you create a PolarDB-X database in the PolarDB-X console, after you select an ApsaraDB RDS for MySQL instance, you can directly import a database in the instance to PolarDB-X for read/write splitting. In this case, you do not need to migrate data, but you also cannot perform horizontal partitioning on tables in the PolarDB-X database.

Support for transactions by read/write splitting

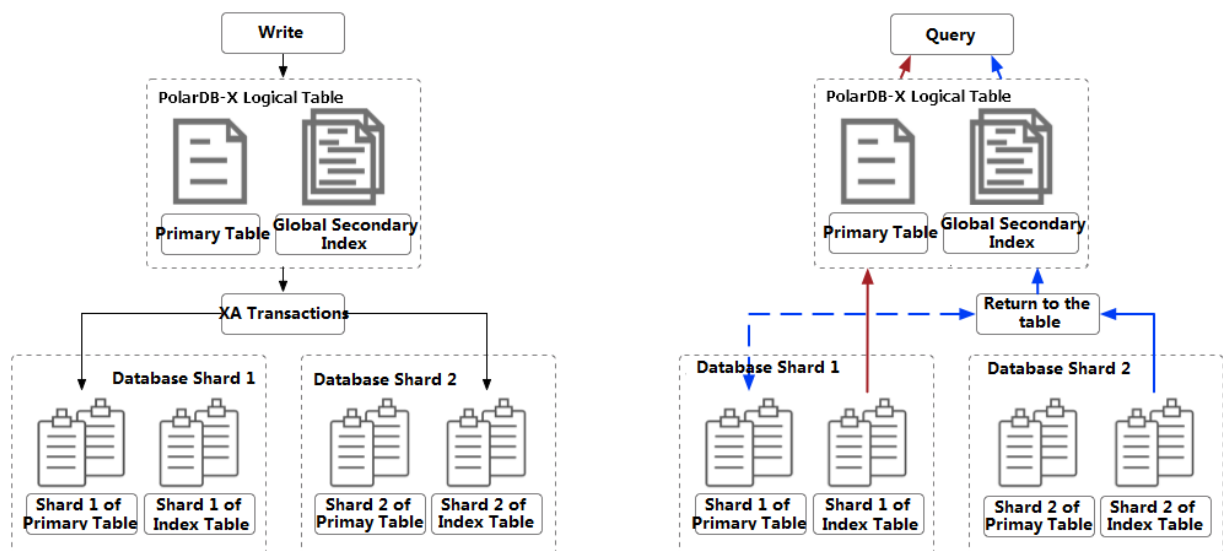
Read/write splitting is valid only for read requests (query requests) that are not in explicit transactions (transactions that need to be explicitly committed or rolled back). Write requests and read requests (including read-only transactions) in explicit transactions are executed in the primary instance and are not distributed to read-only instances.

- Common SQL statements for read requests include SELECT, SHOW, EXPLAIN, and DESCRIBE.
- Common SQL statements for write requests include INSERT, REPLACE, UPDATE, DELETE, and CALL.



9.4.5. Global secondary index

Global secondary indexes of PolarDB-X allow users to add shard dimensions as needed and provides globally unique constraints. Each global secondary index corresponds to an index table and uses XA transactions to ensure strong data consistency between primary tables and index tables.



The global secondary indexes of PolarDB-X provide the following capabilities:

- Add dimensions for sharding.
- Support globally unique indexes.
- Provide XA transactions to ensure strong data consistency between primary tables and index tables.
- Support overwrite columns to reduce overheads from querying the primary table.
- Support Online Schema Change, so the primary table remains unlocked when a global secondary index is added.
- Uses hints to specify indexes to automatically determine whether to query the primary table.

FAQ

Q: What problems can global secondary indexes solve?

A: If the queried dimension is different from the dimension for sharding of a logical table, cross-shard queries are initiated. As cross-shard queries increase, performance problems such as slow query and connection pool exhaustion may occur. Global secondary indexes reduce cross-shard queries and eliminates performance bottlenecks by adding dimensions for sharding. When creating a global secondary index, you need to select a shard key that is different from that of the primary table.

Q: What is the relationship between a global secondary index and a local secondary index?

A:

- A local secondary index stores data rows and corresponding index rows on the same shard in a distributed database. In PolarDB-X, it specifically refers to a MySQL secondary index of a physical table.
- A global secondary index stores data rows and corresponding index rows on different shards, which is different from a local secondary index. A global secondary index quickly determines the data shards involved in the query.
- When PolarDB-X distributes queries to a single shard through a global secondary index, the local secondary index of the shard can improve the performance of the query within the shard.

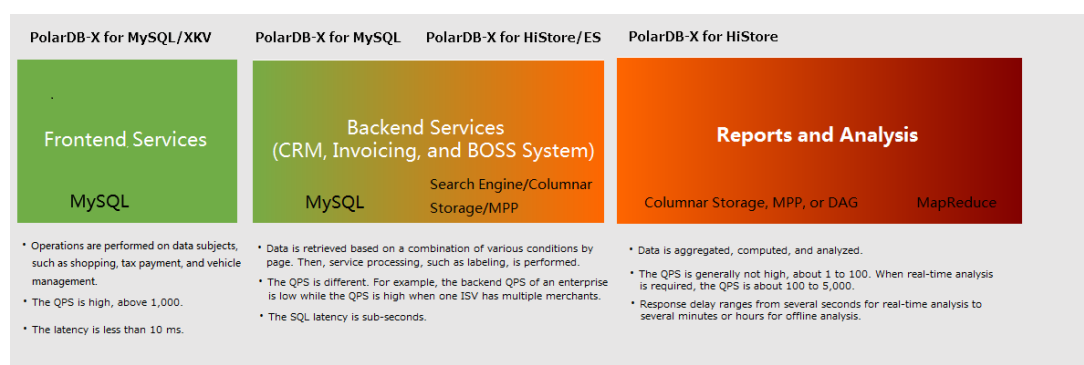
9.5. Scenarios

This topics describes the typical scenarios of PolarDB-X.

PolarDB-X is suitable for businesses that feature high concurrency and low latency in the frontend. It partitions data in specific business scenarios and provides distributed secondary indexes, enabling business databases to keep a high upper limit for queries per second (QPS).

PolarDB-X is trying to support Alibaba columnar databases to meet the needs of the huge-volume storage with low costs, efficient data aggregation, and ad hoc queries.

PolarDB-X scenarios



The following examples are business scenarios for your reference:

- Customer-oriented Internet applications to carry out the business for users (PolarDB-X for MySQL).
- Data businesses that feature high concurrency and low latency in the frontend, such as the bank and hospital counter businesses, Internet of Vehicles (IOV) data operations, tracing, and fuel consumption curves (PolarDB-X for MySQL).
- Storage and aggregation analysis of archived data that is unchangeable (including historical data), such as completed orders, logs, and operation and behavior records (PolarDB-X for HiStore).

9.6. Limits

This topic introduces the restrictions of using PolarDB-X.

Item	Limit
Table shard size	We recommend that a table shard contain a maximum of five million records.
Table shard quantity	Theoretically, the number of table shards in each database shard is not restricted, but depends on the hardware of the PolarDB-X server.
Default database shard quantity for a single ApsaraDB RDS for MySQL instance	8, which cannot be changed.
Distributed JOIN	PolarDB-X supports most JOIN semantics, but also has some restrictions on complex JOIN semantics. For example, JOIN operations between large tables may result in performance or system unavailability due to the high cost and slow speed. Therefore, prevent it whenever possible.

9.7. Terms

This topic defines and analyzes the terms related to PolarDB-X.

Term	Description
PolarDB-X	PolarDB-X is developed by Alibaba. It is a distributed relational databases middleware that is highly compatible with the MySQL protocol and syntax.
PolarDB-XServer (PolarDB-X server node)	PolarDB-X Server is a core component of PolarDB-X. It provides the SQL statement parsing, optimization, routing, and result aggregation functions.
PolarDB-X instance	A PolarDB-X instance is a distributed database server cluster that consists of a group of PolarDB-X server nodes. Each server node is stateless and processes SQL requests.

Term	Description
Specifications of PolarDB-X instances	The specifications of PolarDB-X instances reflect the processing capability of PolarDB-X. Each type provides different CPU and memory resources. Instances with higher specifications provide higher processing capabilities. For example, in a standard PolarDB-X test scenario, an instance with an 8-core CPU and 16 GB of memory has twice the capability of an instance with a 4-core CPU and 8 GB of memory.
Instance upgrade and downgrade	PolarDB-X can adjust the processing capability by upgrading or downgrading instance specifications.
Horizontal partitioning (sharding)	The process that splits a single-instance database into multiple physical database shards, partitions and distributes table data from the single-instance into multiple physical table shards according to sharding rules, and then stores the table shards on different database shards.
Sharding rule	A rule used to partition a logical database table into multiple physical table shards during horizontal partitioning.
Shard key	A database field that generates sharding rules during horizontal partitioning.
Database shard	After the horizontal partitioning of PolarDB-X is complete, data in the logical database is stored in multiple physical storage instances. The physical database in each storage instance is a database shard.
Table shard	After the horizontal partitioning of PolarDB-X is complete, a physical data table in each database shard is called a table shard.
Logical SQL statement	The SQL statement sent by an application to PolarDB-X.
Physical SQL statement	The statement sent to ApsaraDB RDS for MySQL for execution after PolarDB-X parses a logical SQL statement.
Transparent read/write splitting	When a single storage node of PolarDB-X encounters an access bottleneck, you can add read-only instances to share the load on the primary instance. PolarDB-X You do not need to modify application code for the read/write splitting function, so it is called transparent read/write splitting.
Non-partition mode	PolarDB-X supports the extension of database service capabilities through transparent read/write splitting without horizontal partitioning. This is called the non-partition mode.
Smooth scale-out	PolarDB-X can scale out the database by adding storage instance nodes. Smooth scale-out does not affect access to original data.
Broadcast of small tables	PolarDB-X stores tables with small data volumes and infrequent updates in single table mode, which are called small tables. The solution which copies a small table to database shards related to it by JOIN statements through data synchronization to improve the JOIN efficiency, is called broadcast or replication of small tables.

Term	Description
Full table scan	In database partition mode, if no shard key is specified in the SQL statement, PolarDB-X executes the SQL statement on all table shards, merges the results and returns them. This process is called full table scan. To prevent impact on performance, we recommend that you do not perform a full table scan.
PolarDB-X sequence	A PolarDB-X sequence (a 64-digit number of the BIGINT data type in MySQL) aims to ensure that the data (for example, PRIMARY KEY and UNIQUE KEY) in the defined unique field is globally unique and in ordered increments.
PolarDB-X hint (PolarDB-X custom annotations)	A custom hint provided by PolarDB-X to specify certain special actions. It uses related syntax to control the SQL execution to optimize SQL statements.

10. AnalyticDB for PostgreSQL

10.1. What is AnalyticDB for PostgreSQL?

AnalyticDB for PostgreSQL is a distributed analytic database service that uses a massively parallel processing (MPP) architecture in which each instance is composed of multiple compute nodes. AnalyticDB for PostgreSQL provides MPP warehousing services that support horizontal scaling of storage and compute capabilities, online analysis of petabytes of data, and offline processing of Extract, Transform, and Load (ETL) tasks.

AnalyticDB for PostgreSQL is developed based on the PostgreSQL kernel and has the following features:

- Supports the SQL:2003 standard, online analytical processing (OLAP) aggregate functions, views, Procedural Language for SQL (PL/SQL), user-defined functions (UDFs), and triggers. AnalyticDB for PostgreSQL is partially compatible with the Oracle syntax.
- Supports horizontal scaling of storage and compute capabilities based on the MPP architecture. AnalyticDB for PostgreSQL also supports range and list partitioning.
- Supports row store, column store, and multiple indexes. AnalyticDB for PostgreSQL also supports multiple compression methods based on column store to reduce storage costs.
- Supports standard database isolation levels and distributed transactions to ensure data consistency.
- Provides the vector computing engine and the CASCADE-based SQL query optimizer to ensure high-performance SQL analysis.
- Uses a primary/secondary architecture to ensure dual-copy data storage and service availability when O&M and management nodes fail.
- Provides online scaling, system monitoring, and disaster recovery to reduce O&M costs.
- Stores audit logs in Log Service and allows you to download the logs from Log Service. This facilitates long-term storage and management of audit logs.
- Provides automatic fault tolerance for hard disk failures of servers in a cluster, and supports hot swapping of hard disks. In the event of a hard disk failure, services can be restored within 2 minutes.
- Provides permission management and fine-grained audit of user operations.
- Provides a comprehensive permission authentication and isolation mechanism to ensure your data privacy.
- Supports multi-tenancy for resource isolation.
- Supports multi-tenant parallel execution on a cluster by using multiple instances. Tasks from tenants are submitted to queues on different instances for execution. Resources of each AnalyticDB for PostgreSQL instance are isolated among tenants.
- Allows you to configure and manage tenants in a dynamic and centralized manner. You can also isolate resources and query usage statistics of resources. Management of multi-level tenants is supported.
- Supports multi-tenant scheduling of multiple clusters and resource pools.
- Metadatabases support fast switchover for disaster recovery. In the event of a failure, services can be restored within 1 minute.

AnalyticDB for PostgreSQL supports major connection pools such as PgBouncer and pgpool-II.

- PgBouncer: a lightweight connection pool for PostgreSQL.

- Pgpool-II: a connection pool that provides abundant features such as connection pooling, load balancing, and query caching. Pgpool-II also supports automatic retrying of tasks when network exceptions and lock conflicts occur.

10.2. Benefits

This topic describes the benefits of AnalyticDB for PostgreSQL.

- Real-time analysis

Built on the MPP architecture that supports horizontal scaling and can respond to queries on petabytes of data within seconds. AnalyticDB for PostgreSQL supports the leading vector computing feature and intelligent indexes of column store. It also supports the CASCADE-based SQL query optimizer to enable complex queries without the need for tuning.

- Stability and reliability

Provides ACID properties for distributed transactions. Transactions are consistent across nodes and all data is synchronized between primary and secondary nodes. AnalyticDB for PostgreSQL supports distributed deployment and provides transparent monitoring, switching, and restoration to secure your data infrastructure.

- Easy to use

Supports rich SQL syntax and functions, Oracle functions, stored procedures, user-defined functions (UDFs), and isolation levels of transactions and databases. You can use popular business intelligence (BI) software and ETL tools online.

- Ultra-high performance

Supports row store, column store, and multiple indexes. The vector engine provides high-performance analysis and computing capabilities. The CASCADE-based SQL optimizer enables complex queries without the need for tuning. AnalyticDB for PostgreSQL supports high-performance parallel import of data from OSS.

- Flexible scalability

Enables you to scale out compute nodes as well as CPU, memory, and storage resources on demand to improve OLAP performance.

Supports transparent OSS operations. OSS offers a larger storage capacity for cold data that does not require online analysis.

Supports online scaling to add, remove, modify, and query data during data redistribution.

- Resource isolation

Supports multi-tenant parallel execution on a cluster by using multiple instances. Tasks from tenants are submitted to queues on different instances for execution. Resources of each AnalyticDB for PostgreSQL instance are isolated among tenants.

- Permission management

Allows you to configure and manage tenants in a dynamic and centralized manner. You can also isolate resources and query usage statistics of resources. Management of multi-level tenants is supported.

- Resource scheduling

Supports multi-tenant scheduling of multiple clusters and resource pools.

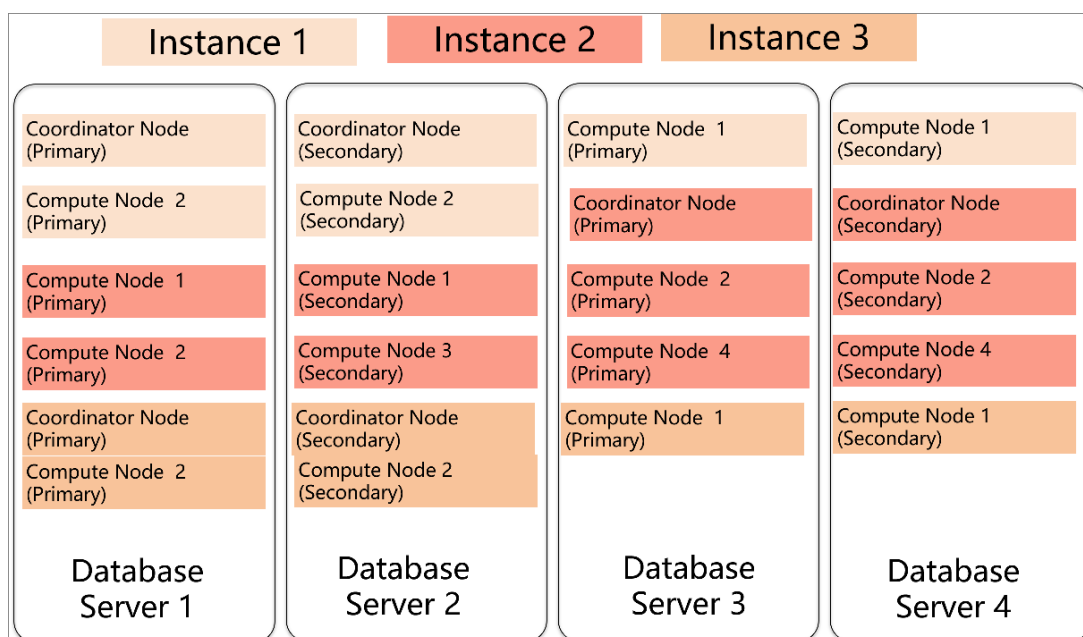
10.3. Architecture

This topic describes the architecture of AnalyticDB for PostgreSQL.

Physical architecture of a cluster

The following figure shows the physical architecture of an AnalyticDB for PostgreSQL cluster.

Physical cluster architecture



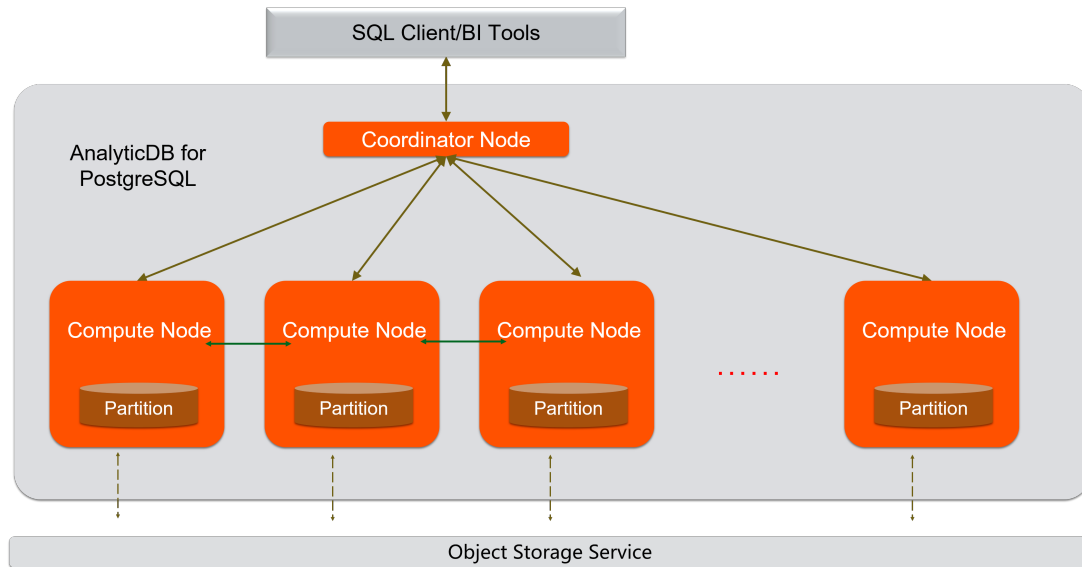
You can create multiple instances in a physical cluster of AnalyticDB for PostgreSQL by using the management and control system. Each instance consists of a coordinator node and multiple compute nodes.

- The coordinator node is used for access from applications. It receives connection requests and SQL query requests from clients and dispatches computing tasks to compute nodes. The cluster deploys a secondary node of the coordinator node on an independent physical server and replicates data from the primary node to the secondary node for failover. The secondary node does not accept external connections.
- Compute nodes are independent instances in AnalyticDB for PostgreSQL. Data is evenly distributed across compute nodes by hash value or RANDOM function, and is analyzed and computed in parallel. Each compute node uses a primary/secondary architecture for automatic failover.

Logical architecture of an instance

You can create multiple instances within an AnalyticDB for PostgreSQL cluster. The following figure shows the logical architecture of an AnalyticDB for PostgreSQL instance.

Logical architecture of an instance



Data is distributed across compute nodes by hash value or RANDOM function of a specified distribution column. Each compute node uses a primary/secondary architecture to ensure dual-copy storage. High-performance network communication is supported across nodes. When the coordinator node receives a request from an application, the coordinator node parses and optimizes SQL statements to generate a distributed execution plan. After the coordinator node sends the execution plan to the compute nodes, the compute nodes perform massively parallel processing of the plan.

10.4. Specifications

This topic describes the specifications of AnalyticDB for PostgreSQL.

AnalyticDB for PostgreSQL supports two storage types: SSD storage and HDD storage. The two storage types provide features that are suited for different scenarios.

- SSD storage: provides excellent I/O capabilities and high analysis performance.
- HDD storage: provides large storage capacity at a low cost.

Specifications

The following table lists the supported specifications of compute nodes in AnalyticDB for PostgreSQL.

Storage type	Number of cores per node	Memory	Available storage space	Total dual-copy storage space	Description
High-performance SSD	1	8GB	80GB	160GB	These specifications are recommended for low-concurrency scenarios that require less than 5 concurrent queries and less than 32 nodes. These specifications are available for 4 to 32 nodes per instance.

Storage type	Number of cores per node	Memory	Available storage space	Total dual-copy storage space	Description
High-performance SSD	4	32GB	320GB	640GB	These specifications are recommended for high-performance SSD storage and available for 8 to 2,048 nodes per instance.
High-capacity HDD	2	16GB	1TB	2TB	These specifications are recommended for low-concurrency scenarios that require less than 5 concurrent queries and less than 8 nodes. These specifications are available for 4 to 32 nodes per instance.
High-capacity HDD	4	32GB	2TB	4TB	These specifications are recommended for high-capacity HDD storage and available for 8 to 2,048 nodes per instance.

10.5. Features

This topic describes the features of AnalyticDB for PostgreSQL.

Distributed architecture

AnalyticDB for PostgreSQL is built on the massively parallel processing (MPP) architecture. Data is evenly distributed across nodes by hash value or RANDOM function, and is analyzed and computed in parallel. Storage and compute capabilities are scaled out by adding nodes to ensure a quick response as the data volume increases.

AnalyticDB for PostgreSQL supports distributed transactions to ensure data consistency among nodes. It supports three transaction isolation levels: `SERIALIZABLE`, `READ COMMITTED`, and `READ UNCOMMITTED`.

High-performance data analysis

AnalyticDB for PostgreSQL supports column store and row store for tables. Row store provides high update performance. Column store provides high online analytical processing (OLAP) aggregate analysis performance for tables. AnalyticDB for PostgreSQL supports B-tree indexes, bit map indexes, and hash indexes to enable high-performance analysis, filtering, and query.

AnalyticDB for PostgreSQL uses the CASCADE-based SQL query optimizer. AnalyticDB for PostgreSQL combines the cost-based optimizer (CBO) with the rule-based optimizer (RBO) to provide SQL optimization features such as automatic subquery decorrelation. These features enable complex queries without the need for tuning.

High-availability service

AnalyticDB for PostgreSQL builds a system for automatic monitoring, diagnostics, and troubleshooting based on the Apsara system. This helps reduce O&M costs.

The coordinator node compiles and optimizes SQL statements by storing database metadata and receiving query requests from clients. The coordinator node uses a primary/secondary architecture to ensure strong consistency of metadata. If the primary coordinator node fails, the service is automatically switched to the secondary coordinator node.

All compute nodes use a primary/secondary architecture to ensure strong data consistency between primary and secondary nodes when data is inserted or updated. If the primary compute node fails, the service is automatically switched to the secondary compute node.

Data synchronization methods and tools

You can use Data Transmission Service (DTS) or DataWorks Data Integration to synchronize data from MySQL or PostgreSQL databases to AnalyticDB for PostgreSQL. You can use popular Extract, Transform, and Load (ETL) tools to import ETL data to and schedule jobs in AnalyticDB for PostgreSQL databases. You can also use standard SQL syntax to query data from formatted files stored in Object Storage Service (OSS) by using foreign tables in real time.

AnalyticDB for PostgreSQL supports popular business intelligence (BI) reporting tools such as Quick BI, DataV, Tableau, and FineReport. AnalyticDB for PostgreSQL also supports ETL tools, including Informatica and Kettle.

High data security

AnalyticDB for PostgreSQL supports the configuration of whitelists. You can add up to 1,000 IP addresses of servers to a whitelist to allow access to your instance and control risks from access sources. AnalyticDB for PostgreSQL also supports Anti-DDoS to monitor inbound traffic in real time. When large amounts of malicious traffic are identified, traffic is scrubbed by filtering requests by IP. If traffic scrubbing is insufficient, blackhole filtering is triggered.

The pgcrypto extension allows you to encrypt columns or tables by using cryptography functions that use encryption algorithms. Algorithms include Message-Digest Algorithm 5 (MD5), Secure Hash Algorithm 1 (SHA-1), SHA-224, SHA-256, SHA-384, SHA-512, Blowfish, Advanced Encryption Standard 128 (AES-128), AES-256, Raw Encryption, Pretty Good Privacy (PGP) symmetric keys, and PGP public keys.

Supported SQL features

- Supports row store and column store.
- Supports multiple indexes, including B-tree indexes, bit map indexes, and hash indexes.
- Supports distributed transactions and standard isolation levels to ensure data consistency among nodes.
- Supports character, date, and arithmetic functions.
- Supports stored procedures, user-defined functions (UDFs), and triggers.
- Supports views.
- Supports range partitioning, list partitioning, and the definition of multi-level partitions.
- Supports partition tables and diverse partition-related operations such as add, drop, rename, clear, swap, and split.
- Supports multiple data types. The following table lists data types and their information.

Data type	Alias	Storage size	Range	Description
bigint	int8	8 bytes	-9223372036854775808 to 9223372036854775807	An integer within a large range.
bigserial	serial8	8 bytes	1 to 9223372036854775807	A large auto-increment integer.
bit [(n)]	None	n bits	A bit string constant	A bit string with a fixed length.
bit varying [(n)]	varbit	Variable	A bit string constant	A bit string with a variable length.
boolean	bool	1 byte	true/false, t/f, yes/no, y/n, 1/0	A Boolean value (true or false).
box	None	32 bytes	((x1,y1),(x2,y2))	A rectangular box on a plane. This data type is not allowed in distribution key columns.
bytea	None	1 byte + binary string	Sequence of octets	A binary string with a variable length.
character [(n)]	char [(n)]	1 byte + n	A string up to n characters in length	A blank-padded string with a fixed length.
character varying [(n)]	varchar [(n)]	1 byte + string size	A string up to n characters in length	A string with a limited variable length.
cidr	None	12 or 24 bytes	None	IPv4 and IPv6 networks.
circle	None	24 bytes	<(x,y),r> (center and radius)	A circle on a plane. This data type is not allowed in distribution key columns.
date	None	4 bytes	4713 BC - 294,277 AD	A calendar date (year, month, day).
decimal [(p, s)]	numeric [(p, s)]	variable	Unlimited	A user-specified precision. This data type is an exact type.
double precision	float8	8 bytes	Precise to 15 decimal digits	A variable precision. This data type is an inexact type.
	float			
inet	None	12 or 24 bytes	None	IPv4 and IPv6 hosts and networks.
integer	int, int4	4 bytes	-2.1E+09 to +2147483647	An integer in typical cases.

Data type	Alias	Storage size	Range	Description
interval [(p)]	None	12 bytes	-178000000 years - 178000000 years	A time range.
json	None	1 byte + json size	JSON string	A string with an unlimited variable length.
lseg	None	32 bytes	((x1,y1),(x2,y2))	A line segment on a plane. This data type is not allowed in distribution key columns.
macaddr	None	6 bytes	None	A MAC address.
money	None	8 bytes	-92233720368547758.08 to +92233720368547758.07	A currency amount.
path	None	16+16n bytes	[(x1,y1),...]	A geometric path on a plane. This data type is not allowed in distribution key columns.
point	None	16 bytes	(x,y)	A geometric point on a plane. This data type is not allowed in distribution key columns.
polygon	None	40+16n bytes	((x1,y1),...)	A closed geometric path on a plane. This data type is not allowed in distribution key columns.
real	float4	4 bytes	Precise to 6 decimal digits	A variable precision. This data type is an inexact type.
serial	serial4	4 bytes	1 to 2147483647	An auto-increment integer.
smallint	int2	2 bytes	-32768 to 32767	An integer within a small range.
text	None	1 byte + string size	A string with a variable length	A string with an unlimited variable length.
time [(p)] [without time zone]	None	8 bytes	00:00:00[.000000] - 24:00:00[.000000]	The time of a day without the time zone.

Data type	Alias	Storage size	Range	Description
time [(p)] with time zone	timetz	12 bytes	00:00:00+1359 - 24:00:00-1359	The time of a day with the time zone.
timestamp [(p)] [without time zone]	None	8 bytes	4713 BC - 294,277 AD	The date and time without the time zone.
timestamp [(p)] with time zone	timestampz	8 bytes	4713 BC - 294,277 AD	The date and time with the time zone.
xml	None	1 byte + xml size	Variable-length XML string	A string with an unlimited variable length.

10.6. Scenarios

This topic describes common scenarios in which AnalyticDB for PostgreSQL can be used.

AnalyticDB for PostgreSQL is applicable to the following online analytical processing (OLAP) data analysis services:

- Extract, Transform, and Load (ETL) for offline data processing

AnalyticDB for PostgreSQL provides the following benefits that make it ideal to optimize complex SQL queries and aggregate and analyze large amounts of data:

- Supports standard SQL syntax, OLAP window functions, and stored procedures.
- Provides the CASCADE-based SQL query optimizer to enable complex queries without the need for tuning.
- Built on the massively parallel processing (MPP) architecture that supports horizontal scaling of storage and compute capabilities to analyze and process petabytes of data.
- Provides column store-based high-performance aggregation of large tables at a high compression ratio to maximize storage capacity.

- Online high-performance query

AnalyticDB for PostgreSQL provides the following benefits for real-time exploration, warehousing, and updating of data:

- Allows you to write and update high-throughput data by performing INSERT, UPDATE, and DELETE operations.
- Allows you to query data based on row store and multiple indexes to obtain results within milliseconds. These indexes include B-tree indexes, bit map indexes, and hash indexes.

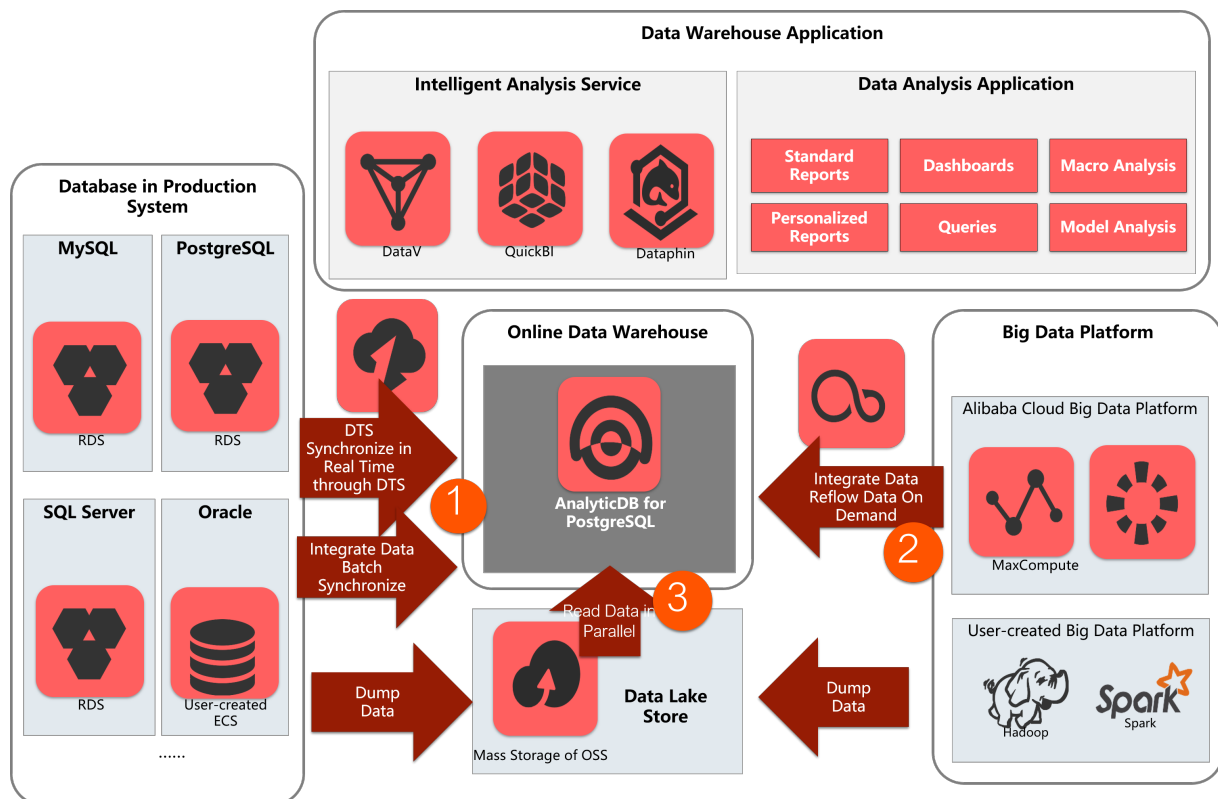
- Supports distributed transactions, standard database isolation levels, and hybrid transaction/analytical processing (HTAP).
- Multi-model data analysis

AnalyticDB for PostgreSQL provides the following benefits to process unstructured data from a variety of sources:

- Supports the PostGIS extension for geographic data analysis and processing.
- Uses the MADlib library of in-database machine learning algorithms to implement AI-native databases.
- Provides high-performance retrieval and analysis of unstructured data such as images, speeches, and texts by means of vector retrieval.
- Supports formats such as JSON and can analyze and process semi-structured data such as logs.

Scenarios

AnalyticDB for PostgreSQL is applicable to the following scenarios:



- Data warehousing service

Data Transmission Service (DTS) can synchronize data in real time in production system databases such as ApsaraDB RDS for MySQL, ApsaraDB RDS for PostgreSQL, and PolarDB as well as traditional databases such as Oracle and SQL Server. Data can also be batch synchronized to AnalyticDB for PostgreSQL by using Data Integration. AnalyticDB for PostgreSQL supports ETL operations on large amounts of data. You can also use DataWorks to schedule these tasks. AnalyticDB for PostgreSQL also provides high-performance online analysis capabilities and can use Quick BI, DataV, Tableau, and FineReport for report presentation and real-time query.

- Big data analytics platform

You can use Data Integration or Object Storage Service (OSS) to import large amounts of data from MaxCompute, Hadoop, and Spark to AnalyticDB for PostgreSQL for high-performance analysis, processing, and exploration.

- Data Lake Analytics (DLA)

AnalyticDB for PostgreSQL can use foreign tables to access large amounts of data stored in OSS in parallel and build an Alibaba Cloud data lake analytics platform.

10.7. Introduction to AnalyticDB for PostgreSQL V6.0

AnalyticDB for PostgreSQL V6.0 is an online database service developed based on the kernel of open source Greenplum Database 6.0. AnalyticDB for PostgreSQL V6.0 improves the real-time processing capabilities for concurrent transactions in data warehouses.

Kernel upgrade

The Greenplum kernel is updated from 4.3 to 6.0, and the PostgreSQL kernel is updated from 8.2 to 9.4. AnalyticDB for PostgreSQL V6.0 provides the following new features:

- JSONB: supports the JSON and JSONB storage types to provide more JSON processing functions and higher processing efficiency.
- UUID: supports the UUID data type.
- GIN and SP-GiST indexes: allow high-performance fuzzy match and retrieval.
- Fine-grained permission control: supports schema-level and column-level permission control and authorization.
- Efficient VACUUM statements: When you execute VACUUM statements to release space, locked pages are skipped and vacuumed at a later time to reduce blocking.
- DBLINK: supports cross-database queries.
- Recursive common table expression (CTE): processes hierarchical or tree-structured data to facilitate multi-level recursive queries.
- PL/SQL enhancement:
 - Supports the RETURN QUERY EXECUTE statement to dynamically execute SQL statements.
 - Supports anonymous blocks.

Hybrid transaction/analytical processing (HTAP) improvement

AnalyticDB for PostgreSQL V6.0 introduces the global deadlock detection mechanism to check and unlock global deadlocks by dynamically collecting and analyzing lock information. Updates and modifications to heap tables can be merely completed by using fine-grained row locks. AnalyticDB for PostgreSQL V6.0 supports concurrent update, delete, and query operations to improve the concurrency and throughput of the system. AnalyticDB for PostgreSQL V6.0 optimized transaction locks to reduce competition for resources when transactions start and end. This way, high-throughput transaction processing is supported in addition to high-performance online analytical processing (OLAP).

OLAP features

- Replicated tables are supported. For dimension tables in data warehouses, replicated tables can be created by using the DISTRIBUTED REPLICATED clause. This reduces data transmission and improves

query efficiency.

- The Zstandard compression algorithm is supported. The compression and decompression performance of this algorithm is over three times better than that of the zlib algorithm.

11.KVStore for Redis

11.1. What is KVStore for Redis?

KVStore for Redis is a database service that is compatible with open source Redis protocols. KVStore for Redis is based on a highly available hot standby architecture and can scale to meet the requirements of high-performance and low-latency read/write operations.

Features

- KVStore for Redis supports various data types, such as strings, lists, sets, sorted sets, hash tables, and streams. This service also supports advanced features, such as transactions, message subscription, and message publishing.
- KVStore for Redis Enhanced Edition (Tair), which is a key-value pair cloud caching service, is an advanced version of KVStore for Redis Community Edition.

Instance type

Instance type	Overview
Community Edition	KVStore for Redis Community Edition is compatible with the data caching service of Redis-native engines. This service supports master-replica instances and cluster instances.
Performance-enhanced instances of KVStore for Redis Enhanced Edition (Tair)	KVStore for Redis Enhanced Edition (Tair) provides a multi-threading model and integrates some features of Alibaba Tair. KVStore for Redis Enhanced Edition (Tair) supports multiple data structures of Tair and is suitable for diverse scenarios.

11.2. Enhanced Edition and supported commands

11.2.1. Performance-enhanced instances of KVStore for Redis Enhanced Edition (Tair)

Performance-enhanced instances of KVStore for Redis Enhanced Edition (Tair) are suitable for scenarios that require high concurrency, high performance, and a large number of reads and writes on hot data. Performance-enhanced instances of KVStore for Redis Enhanced Edition (Tair) support multi-threading and integrate multiple Redis modules.

Benefits

Item	Description

Item	Description
Performance	• Provides read and write performance three times that of Redis-native databases or KVStore for Redis Community Edition with the same specifications. Performance-enhanced instances are suitable for scenarios that require high-frequency read and write requests for hot data. • Responds much faster when processing a large number of queries per second (QPS) compared with Redis-native databases. • Maintains stable performance in high-concurrency scenarios and eliminates connection issues that are caused by traffic spikes during peak hours. • Runs full and incremental synchronization tasks in input/output (I/O) threads to accelerate synchronization.
Enhanced module	Integrates multiple enhanced Redis modules that are developed by Alibaba Cloud. The modules are CAS and CAD commands , TairString commands , TairHash commands , TairGIS commands , TairBloom commands , and TairDoc commands . The enhanced modules provide various solutions, simplify business development in complex scenarios, and allow you to focus on your business development.
Compatibility	Compatible with open source Redis databases. You do not need to modify the code of your application when you use KVStore for Redis.
Scalability	Supports master-replica and cluster architectures. You can scale up or down the specifications of an instance, or upgrade an instance to a cluster instance as needed.

Scenarios

Suitable for scenarios such as live streaming, first-come, first-served events, and online education.

Example:

Issue	Description
Community Edition is not suitable for scenarios that require high queries per second (QPS).	A business system can handle 200,000 QPS or higher for some cached hotkeys. The standard master-replica instances of KVStore for Redis Community Edition cannot maintain high performance during peak hours. Performance-enhanced instances of KVStore for Redis Enhanced Edition (Tair) that use the master-replica architecture can handle requests for popular commodities and provide an excellent user experience. This eliminates performance bottlenecks.
You want to use the current master-replica architecture and improve performance.	Cluster instances have specific limits. Therefore, the current master-replica architecture is retained. Performance-enhanced instances that use the master-replica architecture can improve performance and keep the current architecture unchanged. This eliminates the limits brought by cluster instances after you upgrade the instance to a cluster instance. Therefore, you do not need to adjust your business.

Issue	Description
Self-managed Redis clusters contain a great number of shards, which increase the cost and degrade the performance.	Due to business growth, the number of shards increases. As a result, the management and maintenance costs increase and the performance decreases. Performance-enhanced cluster instances provide high performance and maintain only one third of the number of shards compared with self-managed Redis clusters. This reduces performance loss. KVStore for Redis provides various features to help you manage clusters.

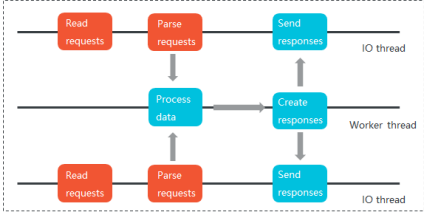
Performance comparison

- Instances of KVStore for Redis Community Edition and open source Redis use a single-threading model. Each data node supports 80,000 to 100,000 QPS.
- KVStore for Redis performance-enhanced instances use a multi-threading model. In this model, I/O threads, worker threads, and auxiliary threads handle requests in parallel. The performance of a data shard of a performance-enhanced instance is three times the performance of a data shard of a Community Edition instance.

The following table describes different scenarios in which various types of instances and architectures are used.

Architecture	Instance type	Description
Standard master-replica instances	Community Edition instances	KVStore for Redis Community Edition instances cannot be used to process more than 100,000 QPS on a single data shard.
	Performance-enhanced instances of KVStore for Redis Enhanced Edition (Tair)	Performance-enhanced instances can be used to process more than 100,000 QPS on a single data shard.
Cluster instances	Community Edition instances	A cluster instance of KVStore for Redis Community Edition contains multiple data shards. Each data shard provides performance similar to that of a master-replica instance. If one of the data shards stores hot data and receives a large number of concurrent requests for hot data, the read/write operations on the other data of this data shard may be delayed. As a result, performance bottlenecks may exist.
	Performance-enhanced instances of KVStore for Redis Enhanced Edition (Tair)	Performance-enhanced instances provide high performance in read/write operations on hot data and reduce maintenance costs.

Threading model comparison

Threading model	Description
Single-threading model 	During the process of request handling, native Redis databases and ApsaraDB for Redis Community Edition instances must undergo the following steps: read requests, parse requests, process data, and then send responses. In this case, network I/O operations and request parsing consume most of the resources that are available.
Multi-threading model 	To increase performance, each performance-enhanced instance of ApsaraDB for Redis runs multiple threads to process the tasks in these steps in parallel. • I/O threads are used to read requests, send responses, and parse commands. • Worker threads are used to process commands and timer events. • Auxiliary threads are used to monitor the statuses of nodes and heartbeats. Each performance-enhanced instance of ApsaraDB for Redis reads and parses requests in I/O threads, places the parsed requests as commands in a queue, and then sends the commands to the worker threads. Then, the worker threads run the commands to process the requests and send the responses to I/O threads by using a different queue. Each performance-enhanced instance of ApsaraDB for Redis supports a maximum of four parallel I/O threads. Unlocked queues and pipelines are used to transmit data between the I/O threads and the worker threads to improve multi-threading performance. Note • The running speeds of threads are accelerated for common data structures, such as string, list, set, hash, zset, hyperloglog, geo, and extension structures. • The replication of API operations such as pub, sub, and blocking is complete in the worker threads and can be accelerated to increase throughput. Performance can be increased by about 50%. • Transactions and Lua scripts require serial execution. No acceleration can be achieved.

Note The multi-threading feature of native Redis 6.0 consumes a large number of CPU resources to deliver performance that is two times higher than the Real Multi-I/O feature of performance-enhanced instances of ApsaraDB for Redis. The Real Multi-I/O feature supports multiple connections, linearly increases in throughput, and provides fully accelerated I/O threads.

11.2.2. CAS and CAD commands

This topic describes the enhanced commands that you can run to process strings on performance-enhanced instances of KVStore for Redis Enhanced Edition. The commands include check-and-set (CAS) and compare-and-delete (CAD).

Prerequisites

The commands for TairHashes that are described in this topic can take effect only if the following conditions are met:

- Performance-enhanced instances of KVStore for Redis Enterprise Edition are used.
- The Redis strings to be managed are stored on the performance-enhanced instance.

Note You can manage Redis strings and TairStrings on a performance-enhanced instance. However, CAS and CAD commands are applicable only to Redis strings.

Commands

Enhanced string commands

Statement	Syntax	Description
CAS	CAS <key> <oldvalue> <newvalue>	Changes the value of a specified key to newvalue if the current value of the key matches the oldvalue parameter. If the current value of the key does not match the oldvalue parameter, the value is not changed. Note The CAS command applies only to Redis strings. To change TairString values, run the EXCAS command.
CAD	CAD <key> <value>	Deletes a specified key if the current value of the key matches the oldvalue parameter. If the current value of the key does not match the oldvalue parameter, the key is not deleted. Note The CAD command applies only to Redis strings. To delete TairString keys, run the EXCAD command.

CAS

- Syntax

CAS <key> <oldvalue> <newvalue>

- Time complexity

O(1)

- Description

This command can be used to change the value of a specified key to a new value if the current value of the key matches a specified value. If the current value of the key does not match the specified value, the value is not changed.

- Parameters and options

Parameter/option	Description
key	The key of the Redis string that you want to manage by using the command.
oldvalue	The value that you compare with the current value of the specified key.
newvalue	Changes the value of the specified key to the value of this parameter if the current value of the key matches the specified value.

- Returned values

- If the operation is successful, a value of 1 is returned.
- If the specified key does not exist, a value of -1 is returned.
- If the operation fails, a value of 0 is returned.
- Otherwise, an error message is returned.

- Example

```
127.0.0.1:6379> SET foo bar
OK
127.0.0.1:6379> CAS foo baa bzz
(integer) 0
127.0.0.1:6379> GET foo
"bar"
127.0.0.1:6379> CAS foo bar bzz
(integer) 1
127.0.0.1:6379> GET foo
"bzz"
```

CAD

- Syntax

CAD <key> <value>

- Time complexity

O(1)

- Description

This command can be used to delete a specified key if the current value of the key matches a specified value. If the current value of the key does not match the specified value, the key is not deleted.

- Parameters and options

Parameter/option	Description
key	The key of the Redis string that you want to manage by using the command.
value	The value that you compare with the current value of the specified key.

- Returned values

- If the operation is successful, a value of 1 is returned.
- If the specified key does not exist, a value of -1 is returned.
- If the operation fails, a value of 0 is returned.
- Otherwise, an error message is returned.

- Example

```
127.0.0.1:6379> SET foo bar
OK
127.0.0.1:6379> CAD foo bzz
(integer) 0
127.0.0.1:6379> CAD not-exists xxx
(integer) -1
127.0.0.1:6379> CAD foo bar
(integer) 1
127.0.0.1:6379> GET foo
(nil)
```

11.2.3. TairString commands

This topic describes the commands that are supported by TairStrings.

Overview

A TairString is a string that includes a version number. Redis-native strings use a key-value pair structure and contain only keys and values. TairStrings contain keys, values, and version numbers. TairStrings can be used in scenarios in which optimistic locking is applied. The **INCRBY** and **INCRBYFLOAT** commands are used to increase or decrease the values of Redis-native strings. You can use TairStrings to limit the range of the results that are returned by the commands. If a result is out of range, an error message is returned.

TairString has the following features:

- A TairString includes a version number.
- TairStrings can be used to limit the range of the results that are returned by the **INCRBY** and **INCRBYFLOAT** commands when you run these commands to increase the values of Redis-native string.

 **Warning** TairStrings are different from Redis-native strings. The commands that are supported by TairStrings and Redis-native strings are not interchangeable.

Prerequisites

The commands for TairHashes take effect only if the following conditions are met:

- Performance-enhanced instances of KVStore for Redis Enterprise Edition are used.
- The TairString to be managed is stored on a performance-enhanced instance.

 **Note** You can manage Redis-native strings and TairStrings on a performance-enhanced instance. However, Redis-native strings do not support the commands that are described in this topic.

Supported commands

TairString commands

Command	Syntax	Overview
EXSET	EXSET <key> <value> [EX time] [PX time] [EXAT time] [PXAT time] [NX XX] [VER version ABS version]	Writes a value to a key.
EXGET	EXGET <key>	Retrieves the value and version number of a TairString.
EXSET VER	EXSETVER <key> <version>	Specifies the version number of a key.
EXINCRBY	EXINCRBY <key> <num> [EX time] [PX time] [EXAT time] [EXAT time] [PXAT time] [NX XX] [VER version ABS version] [MIN minval] [MAX maxval]	Increases or decreases the value of a TairString. The value of the num parameter must be of the long type.
EXINCRBYFLOAT	EXINCRBYFLOAT <key> <num> [EX time] [PX time] [EXAT time] [EXAT time] [PXAT time] [NX XX] [VER version ABS version] [MIN minval] [MAX maxval]	Increases or decreases the value of a TairString. The value of the num parameter must be of the double type.
EXCAS	EXCAS <key> <newvalue> <version>	Changes the value of a specified key when the current version number of the key matches the specified version number. If the update fails, the current value and version number of the key are returned.
EXCAD	EXCAD <key> <version>	Deletes a key when the current version number of the key matches the specified version number. If the operation fails, an error message is returned.
DEL	DEL <key> [key ...]	Deletes one or more TairStrings.

EXSET

- Syntax

EXSET <key> <value> [EX time] [PX time] [EXAT time] [EXAT time] [PXAT time] [NX | XX] [VER version | ABS version]

- Time complexity

O(1)

- Description

This command is used to write a value to a key.

- Parameters and options

Parameter/option	Description
key	The key of the TairString that you want to manage by using the command.
value	The value that you want to write to the specified key.
EX	The relative timeout of the specified key. Unit: seconds. A value of 0 specifies that the key immediately expires.
EXAT	The absolute timeout of the specified key. Unit: seconds. A value of 0 specifies that the key immediately expires.
PX	The relative timeout of the specified key. Unit: milliseconds. A value of 0 specifies that the key immediately expires.
PXAT	The absolute timeout of the specified key. Unit: milliseconds. A value of 0 specifies that the key immediately expires.
NX	Specifies that the value is written to the key only if the specified key does not exist.
XX	Specifies that the value is written to the key only if the specified key exists.
VER	The version number of the specified key. - If the specified key exists, the version number that is specified by this parameter is compared with the current version number. - If the version numbers match, the specified value is written to the key and the version number is increased by 1. - If this parameter does not match the current version number, an error message is returned. - If the specified key does not exist or the current version number of the key is 0, this parameter is ignored. The specified value is written to the key, and the version number is set to 1.
ABS	The absolute version number of the key. Writes the specified value to the key in disregard of the current version number of the key. Then, overwrites the version number with the ABS value.

- Returned values

- If the operation is successful, OK is returned.
- Otherwise, an error message is returned.
- Example

```
127.0.0.1:6379> EXSET foo bar XX
(nil)
127.0.0.1:6379> EXSET foo bar NX
OK
127.0.0.1:6379> EXSET foo bar NX
(nil)
127.0.0.1:6379> EXGET foo
1) "bar"
2) (integer) 1
127.0.0.1:6379> EXSET foo bar1 VER 10
(error) ERR update version is stale
127.0.0.1:6379> EXSET foo bar1 VER 1
OK
127.0.0.1:6379> EXGET foo
1) "bar1"
2) (integer) 2
127.0.0.1:6379> EXSET foo bar2 ABS 100
OK
127.0.0.1:6379> EXGET foo
1) "bar2"
2) (integer) 100
```

EXGET

- Syntax

EXGET <key>
- Time complexity

$O(1)$
- Description

This command is used to retrieve the value and version number of a TairString.
- Parameters and options

key: the key of the TairString that you want to manage.
- Returned values
 - If the operation is successful, the value and version number of the TairString are returned.
 - Otherwise, an error message is returned.
- Example

```
127.0.0.1:6379> EXSET foo bar ABS 100
OK
127.0.0.1:6379> EXGET foo
1) "bar"
2) (integer) 100
127.0.0.1:6379> DEL foo
(integer) 1
127.0.0.1:6379> EXGET foo
(nil)
```

EXSETVER

- Syntax

EXSETVER <key> <version>

- Time complexity

O(1)

- Description

This command is used to specify the version number of a key.

- Parameters and options

Parameter/option	Description
key	The key of the TairString that you want to manage by using the command.
version	The version number that you specify.

- Returned values

- 1: the operation is successful.
- 0: the specified key does not exist.
- Otherwise, an error message is returned.

- Example

```
127.0.0.1:6379> EXSET foo bar
OK
127.0.0.1:6379> EXGET foo
1) "bar"
2) (integer) 1
127.0.0.1:6379> EXSETVER foo 2
(integer) 1
127.0.0.1:6379> EXGET foo
1) "bar"
2) (integer) 2
127.0.0.1:6379> EXSETVER not-exists 0
(integer) 0
```

EXINCRBY

- Syntax

EXINCRBY | EXINCRBY <key> <num> [EX time] [PX time] [EXAT time] [EXAT time] [PXAT time] [NX | XX] [VER version | ABS version] [MIN minval] [MAX maxval]

- Time complexity

$O(1)$

- Description

This command is used to increase or decrease the value of a TairString. The value of the num parameter must be of the long type.

- Parameters and options

Parameter/option	Description
key	The key of the TairString that you want to manage by using the command.
num	The value by which the specified TairString is increased. This value must be an integer.
EX	The relative timeout of the specified key. Unit: seconds. A value of 0 specifies that the key immediately expires.
EXAT	The absolute timeout of the specified key. Unit: seconds. A value of 0 specifies that the key immediately expires.
PX	The relative timeout of the specified key. Unit: milliseconds. A value of 0 specifies that the key immediately expires.
PXAT	The absolute timeout of the specified key. Unit: milliseconds. A value of 0 specifies that the key immediately expires.
NX	Specifies that the value is written to the key only if the specified key does not exist.
XX	Specifies that the value is written to the key only if the specified key exists.
VER	The version number of the specified key. ◦ If the specified key exists, the version number that is specified by this parameter is compared with the current version number. ■ If the version numbers match, the value of the TairString is increased by num and the version number is increased by 1. ■ If this parameter does not match the current version number, an error message is returned. ◦ If the specified key does not exist or the current version number of the key is 0, the specified version number does not take effect. In this case, the TairString value is increased by num and the version number is set to 1.
ABS	The absolute version number of the key. Increases the value of the TairString in disregard of the current version number of the key. Then, overwrites the version number with the ABS value.
MIN	The minimum value of the TairString.
MAX	The maximum value of the TairString.

- Returned values
 - If the operation is successful, the current value of the TairString is returned.
 - Otherwise, an error message is returned.
- Example

```
127.0.0.1:6379> EXINCRBY foo 100
(integer) 100
127.0.0.1:6379> EXINCRBY foo 100 MAX 150
(error) ERR increment or decrement would overflow
127.0.0.1:6379> FLUSHALL
OK
127.0.0.1:6379> EXINCRBY foo 100
(integer) 100
127.0.0.1:6379> EXINCRBY foo 100 MAX 150
(error) ERR increment or decrement would overflow
127.0.0.1:6379> EXINCRBY foo 100 MAX 300
(integer) 200
127.0.0.1:6379> EXINCRBY foo 100 MIN 500
(error) ERR increment or decrement would overflow
127.0.0.1:6379> EXINCRBY foo 100 MIN 500 MAX 100
(error) ERR min or max is specified, but not valid
127.0.0.1:6379> EXINCRBY foo 100 MIN 50
(integer) 300
```

EXINCRBYFLOAT

- Syntax

EXINCRBYFLOAT [EXINCRBYFLOAT <key> <num> [EX time] [PX time] [EXAT time] [EXAT time] [PXAT time] [NX | XX] [VER version | ABS version] [MIN minval] [MAX maxval]]
- Time complexity

O(1)
- Description

This command is used to increase or decrease the value of a TairString. The value of the num parameter must be of the double type.
- Parameters and options

Parameter/option	Description
key	The key of the TairString that you want to manage by using the command.
num	The value by which the specified TairString is increased. The value must be a floating-point number.
EX	The relative timeout of the specified key. Unit: seconds. A value of 0 specifies that the key immediately expires.
EXAT	The absolute timeout of the specified key. Unit: seconds. A value of 0 specifies that the key immediately expires.

Parameter/option	Description
PX	The relative timeout of the specified key. Unit: milliseconds. A value of 0 specifies that the key immediately expires.
PXAT	The absolute timeout of the specified key. Unit: milliseconds. A value of 0 specifies that the key immediately expires.
NX	Specifies that the value is written to the key only if the specified key does not exist.
XX	Specifies that the value is written to the key only if the specified key exists.
VER	The version number of the specified key. ◦ If the specified key exists, the version number that is specified by this parameter is compared with the current version number. ▪ If the version numbers match, the value of the TairString is increased by num and the version number is increased by 1. ▪ If this parameter does not match the current version number, an error message is returned. ◦ If the specified key does not exist or the current version number of the key is 0, the specified version number does not take effect. In this case, the TairString value is increased by num and the version number is set to 1.
ABS	The absolute version number of the key. Increases the value of the TairString in disregard of the current version number of the key. Then, overwrites the version number with the ABS value.
MIN	The minimum value of the TairString.
MAX	The maximum value of the TairString.

- Returned values
 - If the operation is successful, the current value of the TairString is returned.
 - Otherwise, an error message is returned.
- Example

```
127.0.0.1:6379> EXSET foo 100
OK
127.0.0.1:6379> EXINCRBYFLOAT foo 10.123
"110.123"
127.0.0.1:6379> EXINCRBYFLOAT foo 20 MAX 100
(error) ERR increment or decrement would overflow
127.0.0.1:6379> EXINCRBYFLOAT foo 20 MIN 100
"130.123"
127.0.0.1:6379> EXGET foo
1) "130.123"
2) (integer) 3
```

EXCAS

- Syntax

EXCAS <key> <newvalue> <version>

- Time complexity

O(1)

- Description

This command is used to change the version number of a specified key. The version number is changed only if the current version number of the key matches the specified version number.

- Parameters and options

Parameter/option	Description
key	The key of the TairString that you want to manage by using the command.
newvalue	When the current version number of the key matches the specified version number, the specified version number is overwritten by the value of the newvalue parameter.
version	The version number to be compared with the current version number of the specified key.

- Returned values

- If the operation is successful, ["OK", "", version] is returned. The quotation marks ("") represent an empty string, and version represents the current version number of the key.
- If the operation fails, the following error message is returned: ["ERR update version is stale", value, version]. Value represents the current value of the key. Version represents the current version number of the key.
- Otherwise, an error message is returned.

- Example

```
127.0.0.1:6379> EXSET foo bar
OK
127.0.0.1:6379> EXCAS foo bzz 1
1) OK
2)
3) (integer) 2
127.0.0.1:6379> EXGET foo
1) "bzz"
2) (integer) 2
127.0.0.1:6379> EXCAS foo bee 1
1) ERR update version is stale
2) "bzz"
3) (integer) 2
```

EXCAD

- Syntax

EXCAD <key> <version>

- Time complexity

$O(1)$

- Description

This command is used to delete a key when the current version number of the key matches the specified version number.

- Parameters and options

Parameter/option	Description
key	The key of the TairString that you want to manage by using the command.
newvalue	When the current version number of the key matches the specified version number, the specified version number is overwritten by the value of the newvalue parameter.
version	The version number to be compared with the current version number of the specified key.

- Returned values

- 1: the operation is successful.
- -1: the specified key does not exist.
- 0: the operation fails.
- Otherwise, an error message is returned.

- Example

```
127.0.0.1:6379> EXSET foo bar
OK
127.0.0.1:6379> EXGET foo
1) "bar"
2) (integer) 1
127.0.0.1:6379> EXCAD not-exists 1
(integer) -1
127.0.0.1:6379> EXCAD foo 0
(integer) 0
127.0.0.1:6379> EXCAD foo 1
(integer) 1
127.0.0.1:6379> EXGET foo
(nil)
```

11.2.4. TairHash commands

This topic describes the commands supported by a TairHash.

Overview

A TairHash is a hash that allows you to specify the expiration time and version number of a field. TairHashes and Redis-native hashes support multiple commands and provide high performance in data processing. However, Redis-native hashes allow you to specify the expiration time of only keys. TairHashes allow you to specify the expiration time of keys and fields. You can also use TairHashes to specify versions of fields. The improved features of TairHashes allow you to simplify the business development in most scenarios. TairHashes use the active expire algorithm to check the expiration time of fields and delete expired fields. This process does not increase the database response time.

TairHashes have the following features:

- The expiration time and version number for each field can be specified.
- Fields support the active expiration and passive expiration algorithms.
- TairHashes and Redis-native hashes use similar syntax.
- TairHashes support efficient active expiration policies. However, this can increase memory consumption to some extent.



Warning TairHashes are different from Redis-native hashes. The commands that are supported by TairHashes and Redis-native hashes are not interchangeable.

Prerequisites

The following conditions must be met for TairHash commands to take effect:

- **Performance-enhanced instances of KVStore for Redis Enterprise Edition** are used.
- The TairHash to be managed is stored on the performance-enhanced instance.



Note TairHashes and Redis-native hashes are managed on a performance-enhanced instance. The TairHash commands that are described in this topic cannot be applied to Redis-native hashes.

Commands

TairHash commands

Command	Syntax	Description
EXHSET	EXHSET <key> <field> <value> [EX time] [EXAT time] [PX time] [PXAT time] [NX/XX] [VER/ABS version] [NOACTIVE]	Adds a field to a specified TairHash. If the key does not exist, a key for the TairHash is created. If the field has an existing value, this command overwrites the value of the field. When you run this command, the system uses the passive expiration algorithm to delete expired fields.
EXHMSSET	EXHMSSET <key> <field> <value> [field value...]	Sets specified fields to values in a TairHash that matches a specified key. If the key does not exist, a key for the TairHash is created. If the field has an existing value, this command overwrites the value of the field. When you run this command, fields are expired and deleted by using the passive expiration mechanism.

Command	Syntax	Description
EXHPEXPIREAT	EXHPEXPIREAT <key> <field> <milliseconds-timestamp> [VER/ABS version] [NOACTIVE]	Specifies the absolute expiration time of a field in a specified TairHash. Unit: milliseconds. When you run this command, fields are expired and deleted by using the passive expiration mechanism.
EXHPEXPIRE	EXHPEXPIRE <key> <field> <milliseconds> [NOACTIVE]	Specifies the relative expiration time of a field in a TairHash that matches a specified key. Unit: milliseconds. When you run this command, fields are expired and deleted by using the passive expiration mechanism.
EXHEXPIREAT	EXHEXPIREAT <key> <field> <timestamp> [NOACTIVE]	Specifies the absolute expiration time of a field in a specified TairHash. Unit: seconds. When you run this command, fields are expired and deleted by using the passive expiration mechanism.
EXHEXPIRE	EXHEXPIRE <key> <field> <seconds> [NOACTIVE]	Specifies the relative expiration time of a field in a TairHash that matches a specified key. Unit: seconds. When you run this command, fields are expired and deleted by using the passive expiration mechanism.
EXHPTTL	EXHPTTL <key> <field>	Retrieves the remaining expiration time of a field in a specified TairHash. Unit: milliseconds. When you run this command, fields are expired and deleted by using the passive expiration mechanism.
EXHTTL	EXHTTL <key> <field>	Retrieves the remaining expiration time of a field in a specified TairHash. Unit: seconds. When you run this command, fields are expired and deleted by using the passive expiration mechanism.
EXHVER	EXHVER <key> <field>	Retrieves the current version number of a field in a specified TairHash. When you run this command, fields are expired and deleted by using the passive expiration mechanism.
EXHSETVER	EXHSETVER <key> <field> <version>	Sets the version number of a field in a specified TairHash. When you run this command, fields are expired and deleted by using the passive expiration mechanism.

Command	Syntax	Description
EXHINCRBY	EXHINCRBY <key> <field> <num> [EX time] [EXAT time] [PX time] [PXAT time] [VER/ABS version] [MIN minval] [MAX maxval]	Increases the field value in a specified TairHash by an integer. If the specified key does not exist, a TairHash is created. If the specified field does not exist, this command adds the field and sets the value of the field to 0 before a TairHash is created. You can also run the EX, EXAT, PX, or PXAT command to specify the expiration time for the field. When you run this command, fields are expired and deleted by using the passive expiration mechanism. Note To add a field that does not expire, you can run this command without the need to specify an expiration time.
EXHINCRBYFLOAT	EXHINCRBYFLOAT <key> <field> <value> [EX time] [EXAT time] [PX time] [PXAT time] [VER/ABS version] [MIN minval] [MAX maxval]	Increases the value of a field in a specified TairHash by a floating-point number. If the specified key does not exist, a TairHash is created. If the specified field does not exist, this command adds the field and sets the value of the field to 0 before a TairHash is created. You can also run the EX, EXAT, PX, or PXAT command to specify the expiration time for the field. When you run this command, fields are expired and deleted by using the passive expiration mechanism. Note To add a field that does not expire, you can run this command without the need to specify an expiration time.
EXHGET	EXHGET <key> <field>	Retrieves the value of a specified field in a specified TairHash. If the specified key or field does not exist, a value of nil is returned. When you run this command, fields are expired and deleted by using the passive expiration mechanism.
EXHGET WITHVER	EXHGET WITHVER <key> <field>	Retrieves the value and version number of a field in a specified TairHash. If the specified key or field does not exist, a value of nil is returned. When you run this command, fields are expired and deleted by using the passive expiration mechanism.

Command	Syntax	Description
EXHGET	EXHGET <key> <field> [field ...]	Retrieves multiple field values in a specified TairHash in each query if the key of a specified TairHash matches the specified key. If the specified key or field does not exist, a value of nil is returned. When you run this command, fields are expired and deleted by using the passive expiration mechanism.
EXHGETWITHVER	EXHGETWITHVER <key> <field> [field ...]	Retrieves the values and version numbers of multiple fields in a specified TairHash. If the specified key or field does not exist, a value of nil is returned. When you run this command, fields are expired and deleted by using the passive expiration mechanism.
EXHDEL	EXHDEL <key> <field> <field> <field> ...	Deletes a field from a specified TairHash. If the specified key or field does not exist, a value of 0 is returned. If the field is deleted, a value of 1 is returned. When you run this command, fields are expired and deleted by using the passive expiration mechanism.
EXHLEN	EXHLEN <key> [noexp]	Retrieves the number of fields in a specified TairHash. The returned value may include the number of expired fields that are not deleted. If you want to query only the number of fields that are not expired, you can set the <i>noexp</i> parameter.
EXHEXISTS	EXHEXISTS <key> <field>	Checks whether a field exists in a TairHash that matches a specified key. When you run this command, fields are expired and deleted by using the passive expiration mechanism.
EXHSTRLEN	EXHSTRLEN <key> <field>	Retrieves the length of a field value in a specified TairHash. When you run this command, fields are expired and deleted by using the passive expiration mechanism.
EXHKEYS	EXHKEYS <key>	Retrieves all fields in a specified TairHash. Expired fields are filtered out in the returned results. To reduce response time, the system does not delete the expired fields while the system runs the command.
EXHVALS	EXHVALS <key>	Retrieves all field values in a specified TairHash. Expired fields are filtered out in the returned results. To reduce response time, the system does not delete the expired fields while the system runs the command.

Command	Syntax	Description
EXHGET ALL	EXHGET ALL <key>	Retrieves all fields and associated values in a specified TairHash. Expired fields are filtered out in the returned results. To reduce response time, the system does not delete the expired fields while the system runs the command.
EXHSCAN	EXHSCAN <key> <op> <subkey> [MATCH pattern] [COUNT count]	Scans TairHashes that match a specified key. You can set the op parameter to values such as >, >=, <, <=, ==, ^, and \$. This op parameter specifies a scan method. You can also set the MATCH parameter to specify a regular expression and filter out subkeys. The COUNT parameter limits the number of returned values. If you do not set the COUNT parameter, the default value is set to 10. Expired fields are filtered out in the returned results. To reduce response time, the system does not delete the expired fields while the system runs the command.
DEL	DEL <key> [key ...]	Deletes one or more TairHashes.

EXHSET

- Syntax

EXHSET <key> <field> <value> [EX time] [EXAT time] [PX time] [PXAT time] [NX | XX] [VER/ABS version] [NOACTIVE]

- Time complexity

O(1)

- Description

This command is used to add a field to the TairHash that matches a specified key. If the key does not exist, a key for the TairHash is created. If the field has an existing value, this command overwrites the value of the field.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.
value	The value of the specified field. A field can have only one value.
EX	The relative expiration time of the specified field. Unit: seconds. A value of 0 specifies that the field does not expire.

Parameter/option	Description
EXAT	The absolute expiration time of the specified field. Unit: seconds. A value of 0 specifies that the field does not expire.
PX	The relative expiration time of the specified field. Unit: milliseconds. A value of 0 specifies that the field does not expire.
PXAT	The absolute expiration time of the specified field. Unit: milliseconds. A value of 0 specifies that the field does not expire.
NX	Specifies that the value is written only if the field does not exist.
XX	Specifies that the value is written only if the field exists.
VER	The version number of the specified field. ◦ If the specified field exists, the version number specified by this parameter is matched with the current version number: ▪ If the version numbers match, the system continues running the command and increases the version number by 1. ▪ If the version numbers do not match, an error message is returned. ◦ If the specified field does not exist or the current version number of the field is 0, this parameter is ignored. The specified value is written to the field, and then the version number is set to 1.
ABS	The absolute version number of the field. If you set this parameter, the system forcibly writes the specified value to the field regardless of whether the field has a value. Then, the version number is set to the specified ABS value when a field is added.
NOACTIVE	When you set the <i>EX</i> , <i>EXAT</i> , <i>PX</i> , or <i>PXAT</i> parameter, you can set the <i>NOACTIVE</i> parameter to disable the active expiration policy for the field. This allows you to reduce the memory consumption.

- Return values
 - 1: a new field is created and a value is set.
 - 0: the field has a value and the specified value overwrites the current value.
 - -1: the XX parameter is set but the specified field does not exist.
 - -1: the NX parameter is set and the specified field exists.
 - An error message that contains "ERR update version is stale" is returned. The message indicates that the value of the VER parameter does not match the current version number.
 - Otherwise, an exception is returned.

EXHGET

- Syntax

```
EXHGET <key> <field>
```
- Time complexity

$O(1)$

- Description

This command is used to retrieve a value associated with the specified field in a TairHash that matches a specified key.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.

- Return values

- If the field exists and the operation is successful, the value of the field is returned.
- nil: the key or field does not exist.
- Otherwise, an exception is returned.

EXHMMSET

- Syntax

EXHMMSET <key> <field> <value> [field value...]

- Time complexity

O(1)

- Description

This command is used to set specified fields to values in a TairHash that matches a specified key. If the key does not exist, a key for the TairHash is created. If the field has an existing value, this command overwrites the value of the field.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.
value	The value of the specified field. A field can have only one value.

- Return values

- If the operation is successful, OK is returned.
- Otherwise, an exception is returned.

EXHPEXPIREAT

- Syntax

EXHPEXPIREAT <key> <field> <milliseconds-timestamp> [VER/ABS version] [NOACTIVE]

- Time complexity

$O(1)$

- Description

This command is used to specify the absolute expiration time of a field in a TairHash that matches a specified key. Unit: milliseconds.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.
milliseconds-timestamp	The UNIX timestamp. Unit: milliseconds.
VER	The version number of the specified field. ◦ If the specified field exists, the version number specified by this parameter is matched with the current version number: ▪ If the version numbers match, the system continues running the command and increases the version number by 1. ▪ If the version numbers do not match, an error message is returned. ◦ If the specified field does not exist or the current version number of the field is 0, this parameter is ignored. The specified value is written to the field, and then the version number is set to 1.
ABS	The absolute version number of the field. If you set this parameter, the system forcibly writes the specified value to the field regardless of whether the field has a value. Then, the version number is overwritten with the specified ABS value.
NOACTIVE	When you set the <i>EX</i> , <i>EXAT</i> , <i>PX</i> , or <i>PXAT</i> parameter, you can set the <i>NOACTIVE</i> parameter to disable the active expiration policy for the field. This allows you to reduce the memory consumption.

- Return values

- 1: the field exists and a value is set.
- 0: the field does not exist.
- Otherwise, an exception is returned.

EXHPEXPIRE

- Syntax

EXHPEXPIRE <key> <field> <milliseconds> [VER/ABS version] [NOACTIVE]

- Time complexity

O(1)

- Description

This command is used to specify the relative expiration time of a field in a TairHash that matches a specified key. Unit: milliseconds.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.
milliseconds	The relative expiration time of the specified field. Unit: milliseconds.
VER	The version number of the specified field. ◦ If the specified field exists, the version number specified by this parameter is matched with the current version number: ▪ If the version numbers match, the system continues running the command and increases the version number by 1. ▪ If the version numbers do not match, an error message is returned. ◦ If the specified field does not exist or the current version number of the field is 0, this parameter is ignored. The specified value is written to the field, and then the version number is set to 1.
ABS	The absolute version number of the field. If you set this parameter, the system forcibly writes the specified value to the field regardless of whether the field has a value. Then, the version number is overwritten with the specified ABS value.
NOACTIVE	When you set the <i>EX</i> , <i>EXAT</i> , <i>PX</i> , or <i>PXAT</i> parameter, you can set the <i>NOACTIVE</i> parameter to disable the active expiration policy for the field. This allows you to reduce the memory consumption.

- Return values

- 1: the field exists and a value is set.
- 0: the field does not exist.
- Otherwise, an exception is returned.

EXHEXPIREAT

- Syntax

EXHEXPIREAT <key> <field> <timestamp> [VER/ABS version] [NOACTIVE]

- Time complexity

O(1)

- Description

This command is used to specify the absolute expiration time of a field in a TairHash that matches a specified key. Unit: seconds.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.
timestamp	The UNIX timestamp. Unit: seconds.
VER	The version number of the specified field. - If the specified field exists, the version number specified by this parameter is matched with the current version number: - If the version numbers match, the system continues running the command and increases the version number by 1. - If the version numbers do not match, an error message is returned. - If the specified field does not exist or the current version number of the field is 0, this parameter is ignored. The specified value is written to the field, and then the version number is set to 1.
ABS	The absolute version number of the field. If you set this parameter, the system forcibly writes the specified value to the field regardless of whether the field has a value. Then, the version number is overwritten with the specified ABS value.
NOACTIVE	When you set the <i>EX</i> , <i>EXAT</i> , <i>PX</i> , or <i>PXAT</i> parameter, you can set the <i>NOACTIVE</i> parameter to disable the active expiration policy for the field. This allows you to reduce the memory consumption.

- Return values
 - 1: the field exists and a value is set.
 - 0: the field does not exist.
 - Otherwise, an exception is returned.

EXHEXPIRE

- Syntax

```
EXHEXPIRE <key> <field> <seconds>
```
- Time complexity

$O(1)$
- Description

This command is used to specify the relative expiration time of a field in a TairHash that matches a specified key. Unit: seconds.
- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.
seconds	The relative expiration time of the specified field. Unit: seconds.
VER	The version number of the specified field. - If the specified field exists, the version number specified by this parameter is matched with the current version number: - If the version numbers match, the system continues running the command and increases the version number by 1. - If the version numbers do not match, an error message is returned. - If the specified field does not exist or the current version number of the field is 0, this parameter is ignored. The specified value is written to the field, and then the version number is set to 1.
ABS	The absolute version number of the field. If you set this parameter, the system forcibly writes the specified value to the field regardless of whether the field has a value. Then, the version number is overwritten with the specified ABS value.
NOACTIVE	When you set the <i>EX</i> , <i>EXAT</i> , <i>PX</i> , or <i>PXAT</i> parameter, you can set the <i>NOACTIVE</i> parameter to disable the active expiration policy for the field. This allows you to reduce the memory consumption.

- Return values
 - 1: the field exists and a value is set.
 - 0: the field does not exist.
 - Otherwise, an exception is returned.

EXHPTTL

- Syntax

EXHPTTL <key> <field>

- Time complexity

O(1)

- Description

This command is used to retrieve the remaining expiration time of a field in a TairHash that matches a specified key. Unit: milliseconds.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.

Parameter/option	Description
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.

- Return values
 - -2: the specified key or field does not exist.
 - -1: the specified field exists but the TTL value is not specified.
 - The expiration time of the field is returned if the field exists and the expiration time of the field is specified. Unit: milliseconds.
 - Otherwise, an exception is returned.

EXHTTL

- Syntax
EXHTTL <key> <field>
- Time complexity
 $O(1)$
- Description
This command is used to retrieve the remaining expiration time of a field in a TairHash that matches a specified key. Unit: seconds.
- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.

- Return values
 - -2: the specified key or field does not exist.
 - -1: the specified field exists but the TTL value is not specified.
 - The expiration time of the field is returned if the field exists and the expiration time of the field is specified. Unit: seconds.
 - Otherwise, an exception is returned.

EXHVER

- Syntax
EXHVER <key> <field>
- Time complexity
 $O(1)$
- Description

This command is used to retrieve the current version number of a field in a TairHash that matches a specified key.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.

- Return values

- o -1: the specified key does not exist.
- o -2: the specified field does not exist.
- o The version number of the specified field is returned if the operation is successful.
- o Otherwise, an exception is returned.

EXHSETVER

- Syntax

EXHSETVER <key> <field> <version>

- Time complexity

O(1)

- Description

This command is used to set the current version number of a field in a TairHash that matches a specified key.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.

- Return values

- o 0: the specified TairHash or field does not exist.
- o 1: the version number is specified.
- o Otherwise, an exception is returned.

EXHINCRBY

- Syntax

EXHINCRBY <key> <field> <num> [EX time] [EXAT time] [PX time] [PXAT time] [VER/ABS version] [MIN minval] [MAX maxval]

- Time complexity

O(1)

- Description

This command is used to increase the value of a field by num in a TairHash that matches a specified key. The value of the num parameter must be an integer. If the specified TairHash does not exist, a TairHash is created. If the specified field does not exist, this command adds the field and sets the value of the field to 0 before creating a TairHash. When you run this command, the system uses the passive expiration algorithm to delete expired fields.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.
num	The integer by which you want to increase a specified field value.
EX	The relative expiration time of the specified field. Unit: seconds. A value of 0 specifies that the field does not expire.
EXAT	The absolute expiration time of the specified field. Unit: seconds. A value of 0 specifies that the field does not expire.
PX	The relative expiration time of the specified field. Unit: milliseconds. A value of 0 specifies that the field does not expire.
PXAT	The absolute expiration time of the specified field. Unit: milliseconds. A value of 0 specifies that the field does not expire.
VER	The version number of the specified field. - If the specified field exists, the version number specified by this parameter is matched with the current version number: - If the version numbers match, the TairHash is increased by num and the version number is increased by 1. - If the version numbers do not match, an error message is returned. - If the value of the <i>VER</i> parameter is 0, you do not need to check the version number.
ABS	The absolute version number of the field. If you set this parameter, the system forcibly increases the TairHash by num regardless of whether the field has a value. Then, the version number is overwritten with the specified ABS value. The value of this parameter must not be 0.
MIN	The minimum value of the field. If the specified value is smaller than this lower limit, an exception is returned.
MAX	The maximum value of the field. If the specified value is larger than this upper limit, an exception is returned.

Parameter/option	Description
NOACTIVE	When you set the <i>EX</i> , <i>EXAT</i> , <i>PX</i> , or <i>PXAT</i> parameter, you can set the <i>NOACTIVE</i> parameter to disable the active expiration policy for the field. This allows you to reduce the memory consumption.

 **Note** To add a field that does not expire, you can run this command without the need to specify an expiration time.

- Return values
 - The value increased by num is returned if the operation is successful.
 - Otherwise, an exception is returned.

EXHINCRBYFLOAT

- Syntax


```
EXHINCRBYFLOAT <key> <field> <num> [EX time] [EXAT time] [PX time] [PXAT time] [VER/ABS version] [MIN minval] [MAX maxval]
```

- Time complexity

$O(1)$

- Description

This command is used to increase a specified field value by num in a TairHash that matches a specified key. The value of the num parameter must be a floating-point number. If the specified TairHash does not exist, a TairHash is created. If the specified field does not exist, this command adds the field and sets the value of the field to 0 before creating a TairHash. When you run this command, the system uses the passive expiration algorithm to delete expired fields.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.
num	The increment (a floating-point number) to be added to the specified field value.
EX	The relative expiration time of the specified field. Unit: seconds. A value of 0 specifies that the field does not expire.
EXAT	The absolute expiration time of the specified field. Unit: seconds. A value of 0 specifies that the field does not expire.
PX	The relative expiration time of the specified field. Unit: milliseconds. A value of 0 specifies that the field does not expire.
PXAT	The absolute expiration time of the specified field. Unit: milliseconds. A value of 0 specifies that the field does not expire.

Parameter/option	Description
VER	The version number of the specified field. ◦ If the specified field exists, the version number specified by this parameter is matched with the current version number: ▪ If the version numbers match, the TairHash is increased by num and the version number is increased by 1. ▪ If the version numbers do not match, an error message is returned. ◦ If the value of the <i>VER</i> parameter is 0, you do not need to check the version number.
ABS	The absolute version number of the field. If you set this parameter, the system forcibly increases the TairHash by num regardless of whether the field has a value. Then, the version number is overwritten with the specified ABS value. The value of this parameter must not be 0.
MIN	The minimum value of the field. If the specified value is smaller than this lower limit, an exception is returned.
MAX	The maximum value of the field. If the specified value is larger than this upper limit, an exception is returned.
NOACTIVE	When you set the <i>EX</i> , <i>EXAT</i> , <i>PX</i> , or <i>PXAT</i> parameter, you can set the <i>NOACTIVE</i> parameter to disable the active expiration policy for the field. This allows you to reduce the memory consumption.

 **Note** To add a field that does not expire, you can run this command without the need to specify an expiration time.

- Return values
 - The value increased by num is returned if the operation is successful.
 - Otherwise, an exception is returned.

EXHGETWITHVER

- Syntax

```
EXHGETWITHVER <key> <field>
```

- Time complexity

$O(1)$

- Description

This command is used to retrieve the value and version number of a field in a TairHash that matches a specified key.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.

- Return values
 - The value and version number of the field are returned if the field exists and the operation is successful.
 - nil: the key or field does not exist.
 - Otherwise, an exception is returned.

EXHMGET

- Syntax
EXHMGET <key> <field> [field ...]
- Time complexity
O(1)
- Description
This command is used to retrieve multiple field values in a TairHash that matches a specified key.
- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.

- Return values
 - nil: the key does not exist.
 - An array is returned if the specified key and fields exist. Each element in the array corresponds to a field value.
 - An array is returned if the specified key exists but some fields do not exist. Each element in the array corresponds to a field value. The elements of the non-existing fields are displayed as nil.
 - Otherwise, an exception is returned.

EXHMGETWITHVER

- Syntax
EXHMGETWITHVER <key> <field> [field ...]
- Time complexity
O(1)
- Description

This command is used to retrieve the values and version numbers of multiple fields in a TairHash that matches a specified key.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.

- Return values

- nil: the key does not exist.
- An array is returned if the specified key and fields exist. Each element in the array corresponds to a field value and a version number.
- An array is returned if the specified key exists but some fields do not exist. Each element in the array corresponds to a field value and a version number. The elements of the fields that do not exist are displayed as nil.
- Otherwise, an exception is returned.

EXHDEL

- Syntax

EXHDEL <key> <field> <field> <field> ...

- Time complexity

O(1)

- Description

This command is used to delete a field from a TairHash that matches a specified key.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.

- Return values

- 0: the specified key or field does not exist.
- 1: the operation is successful.
- Otherwise, an exception is returned.

EXHLEN

- Syntax

EXHLEN <key> [noexp]

- Time complexity

$O(1)$

- Description

This command is used to retrieve the number of fields in a TairHash that matches a specified key. The returned value may include the number of expired fields that are not deleted.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
noexp	By default, the EXHLEN command does not delete or filter out expired fields. Therefore, the results may include the number of expired fields that are not deleted. If you want to query only the number of fields that are not expired, you can set the <i>noexp</i> parameter. When you set the <i>noexp</i> parameter, ◦ the response time of the EXHLEN command is based on the size of the Tairhash, because the system scans all TairHashes. ◦ The result of the EXHLEN command does not include the number of expired fields that are not deleted.

- Return values

- 0: the specified key or field does not exist.
- The number of fields in the TairHash is returned if the operation is successful.
- Otherwise, an exception is returned.

EXHEXISTS

- Syntax

EXHEXISTS <key> <field>

- Time complexity

$O(1)$

- Description

This command is used to check whether a field exists in a TairHash that matches a specified key.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.

- Return values

- 0: the specified key or field does not exist.
- 1: the specified field exists.
- Otherwise, an exception is returned.

EXHSTRLEN

- Syntax

EXHSTRLEN <key> <field>

- Time complexity

O(1)

- Description

This command is used to retrieve the length of a field value in a TairHash that matches a specified key.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
field	An element of the TairHash. A TairHash key can be mapped to multiple fields.

- Return values

- 0: the specified key or field does not exist.
- The length of the specified field value is returned if the operation is successful.
- Otherwise, an exception is returned.

EXHKEYS

- Syntax

EXHKEYS <key>

- Time complexity

O(1)

- Description

This command is used to retrieve all fields in a TairHash that matches a specified key.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.

- Return values

- If the specified key does not exist, an empty array is returned.
- If the specified key exists, an array is returned. Each element in the array corresponds to a field.
- Otherwise, an exception is returned.

EXHVALS

- Syntax

EXHVALS <key>

- Time complexity

$O(1)$

- Description

This command is used to retrieve all field values in a TairHash that matches a specified key.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.

- Return values

- If the specified key does not exist, an empty array is returned.
- If the specified key exists, an array is returned. Each element in the array corresponds to a field value.
- Otherwise, an exception is returned.

EXHGETALL

- Syntax

EXHGETALL <key>

- Time complexity

$O(1)$

- Description

This command is used to retrieve all fields and their values in a TairHash that matches a specified key.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.

- Return values

- If the specified key does not exist, an empty array is returned.
- If the specified key exists, an array is returned. Each element in the array corresponds to a field-value pair.
- Otherwise, an exception is returned.

EXHSCAN

- Syntax

EXHSCAN <key> <op> <subkey> [MATCH pattern] [COUNT count]

- Time complexity

$O(1)$ and $O(N)$

- Description

This command is used to scan TairHashes that match a specified key.

- Parameters and options

Parameter/option	Description
key	The key of the TairHash that you want to manage.
op	The position from which a scan starts. Valid values: ◦ >: specifies that the scan starts from the first field with the value of the key greater than the subkey. ◦ >=: specifies that the scan starts from the first field with the value of the key greater than or equal to the subkey. ◦ <: specifies that the scan starts from the first field with the value of the key smaller than the subkey. ◦ <=: specifies that the scan starts from the first field with the value of the key smaller than or equal to the subkey. ◦ ==: specifies that the scan starts from the first field with the value of the key equal to the subkey. ◦ ^: specifies that the scan starts from the first field. ◦ \$: specifies that the scan starts from the last field.
subkey	Specifies the position from which a scan starts. This parameter is set together with the op parameter. If op is set to ^ or \$, this parameter does not take effect.
MATCH	The criteria used to filter the scanning result.

- Return values

- If the specified key does not exist, an empty array is returned.
- If the specified key exists, an array is returned. Each element in the array corresponds to a field-value pair.
- Otherwise, an exception is returned.

11.2.5. TairGIS commands

TairGIS is a data structure that uses R-tree indexes and supports the API operations for Geographic Information System (GIS). Compared with Redis GEO commands, which allow you to use GeoHash and Redis Sorted Set to query points, TairGIS allows you to query points, lines, and planes, and provides more features.

Prerequisites

- Performance-enhanced instances of KVStore for Redis Enterprise Edition are used.
- The KVStore for Redis instance is upgraded to the latest minor version. For more information about how to upgrade the minor version of a KVStore for Redis instance, see *Upgrade the minor version in User Guide*.

Features

- Supports R-tree indexing.
- Allows you to query points, lines, and planes. This includes queries for the intersection of sets.
- Compatible with Redis-native GEO commands.

Commands

TairGIS commands

Statement	Syntax	Description
GIS.ADD	<pre>GIS.ADD <area> <polygonName> <polygonWkt> [<polygonName> <polygonWkt> ...]</pre>	Adds one or more specified polygons to a specified area. The polygons are described in well-known text (WKT).  Note WKT is a text markup language that you can use to represent vector geometry objects on a map and the spatial reference systems of spatial objects. WKT also allows you to perform transformations between spatial reference systems.
GIS.GET	<pre>GIS.GET <area> <polygonName></pre>	Retrieves the WKT information about a specified polygon in an area.
GIS.GET ALL	<pre>GIS.GETALL <area> [WITHOUTWKT]</pre>	Queries all polygons in a specified area. The names and WKT information of the polygons are returned.
GIS.DEL	<pre>GIS.DEL <area> <polygonName></pre>	Deletes a specified polygon in an area.
DEL	<pre>DEL <key> [key ...]</pre>	Deletes one or more TairGIS data structures. This is a Redis-native command.
GIS.CONTAINS	<pre>GIS.CONTAINS <area> <polygonWkt> [WITHOUTWKT]</pre>	Checks whether a polygon in a specified area consists of a specified point, line, or plane.
GIS.INTERSECT S	<pre>GIS.INTERSECTS <area> <polygonWkt></pre>	Queries the intersection relationship between a polygon in a specified area and a specified point, line, or plane.
GIS.SEARCH	<pre>GIS.SEARCH [RADIUS longitude latitude distance m km ft mi] [MEMBER field distance m km ft mi] [GEOID geom] [COUNT count] [ASC DESC] [WITHDIST] [WITHOUTWKT]</pre>	Retrieves points within a sphere whose radius, latitude, and longitude are specified.
GIS.WITHIN	<pre>GIS.WITHIN <area> <polygonWkt> [WITHOUTWKT]</pre>	Retrieves points, lines, or planes within a specified polygon in an area.

Parameters

Parameter	Description
area	The geometric area in which you want to manage the data.
PolygonName	The name of the polygon that you want to manage.
polygonWkt	The description of a polygon, which is described by using WKT. The following types are supported: - POINT: the WKT information that describes a point, such as <code>'POINT (30 11)'</code>. - LINESTRING: the WKT information that describes a line, such as <code>'LINESTRING (30 10, 40 40)'</code>. - POLYGON: the WKT information that describes a polygon, such as <code>'POLYGON ((31 20, 29 20, 29 21, 31 31))'</code>.  Note MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, GEOMETRY, and COLLECTION are not supported.
WITHOUTWKT	Specifies whether to return the WKT information of polygons. If you run the <code>GIS.GET ALL</code> , <code>GIS.CONTAINS</code> , <code>GIS.SEARCH</code> , or <code>GIS.WITHIN</code> command and specify the <code>WITHOUTWKT</code> parameter, the WKT information of the polygon is not returned.

GIS.ADD

- Syntax

```
GIS.ADD <area> <polygonName> <polygonWkt> [<polygonName> <polygonWkt>...]
```

- Time complexity, which is used to indicate the trend of statement execution time as the data size increases.

$O(\log n)$

- Description

This command is used to add one or more polygons to a specified area. The polygons are described in WKT.

- Returned values

- If the operation is successful, the number of successful inserts and updates are returned.
- Otherwise, an exception is returned.

- Example

```
127.0.0.1:6379> GIS.ADD hangzhou campus 'POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))'
(integer) 1
```

GIS.GET

- Syntax

```
GIS.GET <area> <polygonName>
```

- Time complexity

$O(1)$

- Description

This command is used to retrieve the WKT information about a specified polygon in an area.

- Returned values

- If the operation is successful, the WKT information about the polygon is returned.
- If the specified area or polygon name does not exist, a value of nil is returned.
- Otherwise, an exception is returned.

- Example

```
127.0.0.1:6379> GIS.ADD hangzhou campus 'POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))'  
(integer) 1  
127.0.0.1:6379> GIS.GET hangzhou campus  
"POLYGON((30 10,40 40,20 40,10 20,30 10))"  
127.0.0.1:6379> GIS.GET hangzhou not-exists  
(nil)  
127.0.0.1:6379> GIS.GET not-exists campus  
(nil)
```

GIS.GETALL

- Syntax

```
GIS.GETALL <area> [WITHOUTWKT]
```

- Time complexity

$O(n)$

- Description

This command is used to query all polygons in a specified area. The names and WKT information of the polygons are returned. If you specify the *WITHOUTWKT* parameter, only the name of the polygon is returned.

- Returned values

- If the operation is successful, the name and WKT information of the polygon are returned. If you specify the *WITHOUTWKT* parameter, only the name of the polygon is returned.
- If no data is found, a value of nil is returned.
- Otherwise, an exception is returned.

- Example

```
127.0.0.1:6379> GIS.ADD hangzhou campus 'POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))'  
(integer) 1  
127.0.0.1:6379> GIS.GETALL hangzhou  
1) "campus"  
2) "POLYGON((30 10,40 40,20 40,10 20,30 10))"  
127.0.0.1:6379> GIS.GETALL hangzhou WITHOUTWKT  
1) "campus"
```

GIS.DEL

- Syntax

GIS.DEL <area> <polygonName>

- Time complexity

$O(\log n)$

- Description

This command is used to delete a specified polygon in an area.

- Returned values

- If the operation is successful, OK is returned.
- If the specified area or polygon name does not exist, a value of nil is returned.
- Otherwise, an exception is returned.

- Example

```
127.0.0.1:6379> GIS.ADD hangzhou campus 'POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))'
(integer) 1
127.0.0.1:6379> GIS.GET hangzhou campus
"POLYGON((30 10,40 40,20 40,10 20,30 10))"
127.0.0.1:6379> GIS.DEL hangzhou not-exists
(nil)
127.0.0.1:6379> GIS.DEL not-exists campus
(nil)
127.0.0.1:6379> GIS.DEL hangzhou campus
OK
127.0.0.1:6379> GIS.GET hangzhou campus
(nil)
```

GIS.CONTAINS

- Syntax

GIS.CONTAINS <area> <polygonWkt>

- Time complexity

- Optimal time complexity: $O(\log_M n)$
- Least desirable time complexity: $\log(n)$.

- Description

This command is used to check whether a polygon in a specified area contains a specified point, line, or plane.

- Returned values

- If the operation is successful, the name and WKT information of the specified polygon that contains the specified point, line, or plane are returned. If you specify the *WITHOUT WKT* parameter, only the name of the polygon is returned.
- If no data is found, a value of nil is returned.
- Otherwise, an exception is returned.

- Example

```

127.0.0.1:6379> GIS.ADD hangzhou campus 'POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))'
(integer) 1
127.0.0.1:6379> GIS.CONTAINS hangzhou 'POINT (30 11)'
1) "0"
2) 1) "campus"
   2) "POLYGON((30 10,40 40,20 40,10 20,30 10))"
127.0.0.1:6379> GIS.CONTAINS hangzhou 'LINESTRING (30 10, 40 40)'
1) "0"
2) 1) "campus"
   2) "POLYGON((30 10,40 40,20 40,10 20,30 10))"
127.0.0.1:6379> GIS.CONTAINS hangzhou 'POLYGON ((31 20, 29 20, 29 21, 31 31))'
1) "0"
2) 1) "campus"
   2) "POLYGON((30 10,40 40,20 40,10 20,30 10))"

```

GIS.INTERSECTS

- Syntax

GIS.INTERSECTS <area> <polygonWkt>

- Time complexity

- Optimal time complexity: $O(\log_M n)$
- Least desirable time complexity: $\log(n)$.

- Description

This command is used to query the intersection relationship between a polygon in a specified area and a specified point, line, or plane.

- Returned values

- If the operation is successful, the name and WKT information of the specified polygon that intersects with the specified point, line, or plane are returned.
- If no data is found, a value of nil is returned.
- Otherwise, an exception is returned.

- Example

```

127.0.0.1:6379> GIS.ADD hangzhou campus 'POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))'
(integer) 1
127.0.0.1:6379> GIS.INTERSECTS hangzhou 'POINT (30 11)'
1) "0"
2) 1) "campus"
   2) "POLYGON((30 10,40 40,20 40,10 20,30 10))"
127.0.0.1:6379> GIS.INTERSECTS hangzhou 'LINESTRING (30 10, 40 40)'
1) "0"
2) 1) "campus"
   2) "POLYGON((30 10,40 40,20 40,10 20,30 10))"
127.0.0.1:6379> GIS.INTERSECTS hangzhou 'POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))'
1) "0"
2) 1) "campus"
   2) "POLYGON((30 10,40 40,20 40,10 20,30 10))"
127.0.0.1:6379>

```

GIS.SEARCH

- Syntax

```
GIS.SEARCH [RADIUS longitude latitude distance m|km|ft|mi]
[MEMBER field distance m|km|ft|mi]
[GEOM geom]
[COUNT count]
[ASC|DESC]
[WITHDIST]
[WITHOUTWKT]
```

- Time complexity

- Optimal time complexity: $O(\log_M n)$
- Least desirable time complexity: $\log(n)$.

- Description

This command is used to retrieve points within a sphere whose radius, latitude, and longitude are specified. The following parameters are supported:

- *RADIUS*: searches points by the longitude, latitude, and radius. Specify the parameter values in the following order: longitude, latitude, radius, and radius unit, such as `RADIUS 15 37 200 km`.
- *MEMBER*: searches points by the latitude and longitude from an existing polygon, and a specified radius. Specify the parameter values in the following order: polygon name, radius, and radius unit, such as, `MEMBER Agrigento 100 km`.
- *GEOM*: specifies the search range in the WKT format for a random polygon, such as `GIS.SEARCH Sicily "POINT (13.361389 38.115556)"`.
- *COUNT*: limits the number of returned entries, such as `COUNT 3`.
- *ASC|DESC*: sorts returned entries by distance. For example, ASC indicates that the entries are sorted from nearest to farthest from the center.
- *WITHDIST*: specifies whether to return the distance.
- *WITHOUTWKT*: specifies whether to return the WKT information of polygons.

- Returned values

- If the operation is successful, the name and WKT information of the specified polygon are returned.
- If no data is found, a value of nil is returned.
- Otherwise, an exception is returned.

- Example

```

127.0.0.1:6379> GIS.ADD Sicily "Palermo" "POINT (13.361389 38.115556)" "Catania" "POINT(15.087269 37.502669)"
(integer) 2
127.0.0.1:6379> GIS.SEARCH Sicily RADIUS 15 37 200 km WITHDIST
1) "2"
2) 1) "Palermo"
   2) "POINT(13.361389 38.115556)"
   3) "190.4424"
   4) "Catania"
   5) "POINT(15.087269 37.502669)"
   6) "56.4413"
127.0.0.1:6379> GIS.SEARCH Sicily RADIUS 15 37 200 km WITHDIST WITHOUTWKT
1) "2"
2) 1) "Palermo"
   2) "190.4424"
   3) "Catania"
   4) "56.4413"
127.0.0.1:6379> GIS.SEARCH Sicily RADIUS 15 37 200 km WITHDIST WITHOUTWKT ASC
1) "2"
2) 1) "Catania"
   2) "56.4413"
   3) "Palermo"
   4) "190.4424"
127.0.0.1:6379> GIS.SEARCH Sicily RADIUS 15 37 200 km WITHDIST WITHOUTWKT ASC COUNT 1
1) "2"
2) 1) "Catania"
   2) "56.4413"
127.0.0.1:6379> GIS.ADD Sicily "Agrigento" "POINT (13.583333 37.316667)"
(integer) 1
127.0.0.1:6379> GIS.SEARCH Sicily MEMBER Agrigento 100 km
1) "2"
2) 1) "Palermo"
   2) "POINT(13.361389 38.115556)"
   3) "Agrigento"
   4) "POINT(13.583333 37.316667)"

```

GIS.WITHIN

- Syntax

```
GIS.WITHIN <area> <polygonWkt> [WITHOUTWKT]
```

- Time complexity

- Optimal time complexity: $O(\log_M n)$
- Least desirable time complexity: $\log(n)$.

- Description

This command is used to retrieve points, lines, or planes within a specified polygon in an area. If you specify the *WITHOUTWKT* parameter, only the name of the polygon is returned.

- Returned values

- If the operation is successful, the name and WKT information of the polygon are returned.

- If no data is found, a value of nil is returned.
- Otherwise, an exception is returned.
- Example

```
127.0.0.1:6379> GIS.ADD hangzhou campus 'POINT (30 10)'
(integer) 1
127.0.0.1:6379> GIS.ADD hangzhou campus1 'LINESTRING (30 10, 40 40)'
(integer) 1
127.0.0.1:6379> GIS.WITHIN hangzhou 'POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))'
1) "2"
2) 1) "campus"
   2) "POINT(30 10)"
   3) "campus1"
   4) "LINESTRING(30 10,40 40)"
```

11.2.6. TairBloom commands

This topic describes the commands that are supported by a TairBloom.

Overview

TairBloom is a Bloom filter that supports dynamic scaling. TairBloom is a space-efficient probabilistic data structure that consumes minimal memory to check whether an element exists. TairBloom supports dynamic scaling and maintains a stable false positive rate during scaling.

You can use bitmaps on Redis data structures, such as hashes, sets, and strings, to implement similar features of TairBloom. However, these data structures may consume a large amount of memory or fail to maintain a stable false positive rate during dynamic scaling. You can use TairBloom to check whether large volumes of data exist. In this case, a specific false positive rate is allowed. You can use the built-in Bloom filter of TairBloom without further development or the need to create an extra Bloom filter.

Key features:

- Consumes minimal memory.
- Enables dynamic scaling.
- Maintains a stable custom false positive rate during scaling.

Prerequisites

The commands for TairBloom take effect only if the following conditions are met:

- **Performance-enhanced instances of KVStore for Redis Enterprise Edition** are used.
- The TairBloom that you want to manage is stored on a performance-enhanced instance.

Commands

TairBloom commands

Command	Syntax	Description
BF.RESERVE	BF.RESERVE <key> <error_rate> <capacity>	Creates an empty TairBloom filter with a specific capacity. The error_rate parameter specifies the false positive rate of the TairBloom filter.

Command	Syntax	Description
BF.ADD	BF.ADD <key> <item>	Adds an item to a specified TairBloom filter.
BF.MADD	BF.MADD <key> <item> [item...]	Adds multiple items to a specified TairBloom filter.
BF.EXISTS	BF.EXISTS <key> <item>	Checks whether an item exists in a specified TairBloom filter.
BF.MEXISTS	BF.MEXISTS <key> <item> [item...]	Checks whether multiple items exist in a specified TairBloom filter.
BF.INSERT	BF.INSERT <key> [CAPACITY cap] [ERROR error] [NOCREATE] ITEMS <item...>	Adds multiple items to a specified TairBloom filter. If the TairBloom filter does not exist, you can specify whether to create a TairBloom filter. You can also specify the capacity and false positive rate of the new TairBloom filter.
BF.DEBUG	BF.DEBUG <key>	Retrieves the information about a specified TairBloom filter. The information includes the number of layers, the number of items at each layer, and the false positive rate.
DEL	DEL <key> [key ...]	Deletes one or more TairBlooms.  Note You cannot delete the items that are added to a TairBloom. You can run the DEL command to delete the TairBloom.

BF.RESERVE

- Syntax

BF.RESERVE <key> <error_rate> <capacity>

- Time complexity

O(1)

- Description

This command is used to create an empty TairBloom filter with a specific capacity. The error_rate parameter specifies the false positive rate of the TairBloom filter.

- Parameters and options

Parameter/option	Description
key	The key of the TairBloom filter that you want to manage.
error_rate	The expected false positive rate. The value of this parameter must be between 0 and 1. A lower value indicates higher memory usage and CPU utilization of the TairBloom filter.

Parameter/option	Description
capacity	The initial capacity of the TairBloom filter. This parameter specifies the maximum number of items that can be added to the TairBloom filter. If the number of items that are added to the TairBloom filter exceeds the specified capacity, TairBloom expands the capacity by increasing the layers of the Bloom filter. During the scaling process, the number of items in the Tairbloom filter exponentially increases and the performance linearly decreases. To query a specific item after a layer is added to the filter, TairBloom may iterate through multiple layers of the filter. The capacity of each new layer is twice that of the previous layer. If your workloads require high performance, we recommend that you add items to TairBloom based on your business requirements to avoid automatic scaling.

- Returned values
 - If the operation is successful, OK is returned.
 - If the operation is not successful, an error message is returned.

BF.ADD

- Syntax


```
BF.ADD <key> <item>
```
- Time complexity

$O(\log N)$. N specifies the number of layers of the TairBloom filter.
- Description

This command is used to add an item to a specified TairBloom filter.
- Parameters and options

Parameter/option	Description
key	The key of the TairBloom filter that you want to manage.
item	The item that you want to add to the TairBloom filter.

- Returned values
 - 1: The item does not exist in the filter.
 - 0: The item may exist in the filter.
 - If the operation is not successful, an error message is returned.

BF.MADD

- Syntax


```
BF.MADD <key> <item> [item..]
```
- Time complexity

$O(\log N)$. N specifies the number of layers of the TairBloom filter.

- Description

This command is used to add multiple items to a specified TairBloom filter.

- Parameters and options

Parameter/option	Description
key	The key of the TairBloom filter that you want to manage.
item	The items that you want to add to the TairBloom filter. You can specify multiple items.

- Returned values

- If the operation is successful, an array is returned. The values in the returned array can be 1 or 0. If a specified item does not exist, the value is 1. If a specified item may exist, the value is 0.
- If the operation is not successful, an error message is returned.

BF.EXISTS

- Syntax

BF.EXISTS <key> <item>

- Time complexity

$O(\log N)$. N specifies the number of layers of the TairBloom filter.

- Description

This command is used to check whether an item exists in a specified TairBloom filter.

- Parameters and options

Parameter/option	Description
key	The key of the TairBloom filter that you want to manage.
item	The item that you want to query in the TairBloom filter.

- Returned values

- 0: The specified item does not exist in the filter.
- 1: The specified item may exist in the filter.
- If the operation is not successful, an error message is returned.

BF.MEXISTS

- Syntax

BF.MEXISTS <key> <item> [item...]

- Time complexity

$O(\log N)$. N specifies the number of layers of the TairBloom filter.

- Description

This command is used to check whether multiple items exist in a specified TairBloom filter.

- Parameters and options

Parameter/option	Description
key	The key of the TairBloom filter that you want to manage.
item	The items that you want to query in the TairBloom filter. You can specify multiple items.

- Returned values

- If the operation is successful, an array is returned. The values in the returned array can be 1 or 0. If a specified item does not exist, the value is 0. If a specified item may exist, the value is 1.
- If the operation is not successful, an error message is returned.

BF.INSERT

- Syntax

BF.INSERT <key> [CAPACITY cap] [ERROR error] [NOCREATE] ITEMS <item...>

- Time complexity

$O(\log N)$. N specifies the number of layers of the TairBloom filter.

- Description

This command is used to add multiple items to a specified TairBloom filter. If the TairBloom filter does not exist, you can specify whether to create a TairBloom filter. You can also specify the capacity and false positive rate of the new TairBloom filter.

- Parameters and options

Parameter/option	Description
key	The key of the TairBloom filter that you want to manage.
CAPACITY	The initial capacity of the TairBloom filter. This parameter specifies the maximum number of items that can be added to the TairBloom filter. If the filter exists, you do not need to specify this parameter. If the number of items that are added to the TairBloom filter exceeds the specified capacity, TairBloom expands the capacity by increasing the layers of the Bloom filter. During the scaling process, the number of items in the Tairbloom filter exponentially increases and the performance linearly decreases. To query a specific item after a layer is added to the filter, TairBloom may iterate through multiple layers of the filter. The capacity of each new layer is twice that of the previous layer. If your workloads require high performance, we recommend that you add items to TairBloom based on your business requirements to avoid automatic scaling.
ERROR	The expected false positive rate. If the filter exists, you do not need to specify this parameter. The value of this parameter must be between 0 and 1. A lower value indicates higher memory usage and CPU utilization of the TairBloom filter.

Parameter/option	Description
NOCREATE	Specifies that the specified TairBloom filter is not automatically created if the filter does not exist. This parameter cannot be specified together with CAPACITY or ERROR.
ITEMS	All items that you want to add to the TairBloom filter.

- Returned values
 - If the operation is successful, an array is returned. The values in the returned array can be 1 or 0. If a specified item does not exist, the value is 1. If a specified item may exist, the value is 0.
 - If the operation is not successful, an error message is returned.

BF.DEBUG

- Syntax
BF.DEBUG <key>
- Time complexity
 $O(\log N)$. N specifies the number of layers of the TairBloom filter.

- Description

This command is used to retrieve the information about a specific TairBloom filter. The information includes the number of layers, the number of items at each layer, and the false positive rate.

- Parameters and options

Parameter/option	Description
key	The key of the TairBloom filter that you want to manage.

- Returned values
 - If the operation is successful, an array is returned. The values in the returned array can be 1 or 0. If a specified item does not exist, the value is 1. If a specified item may exist, the value is 0.
 - If the operation is not successful, an error message is returned.

Memory usage test result

Capacity (number of elements)	false positive:0.01	false positive:0.001	false positive:0.0001
100000	0.12 MB	0.25 MB	0.25 MB
1000000	2 MB	2 MB	4 MB
10000000	16 MB	32 MB	32 MB
100000000	128 MB	256 MB	256 MB
1000000000	2 GB	2 GB	4 GB

11.2.7. TairDoc commands

This topic describes the commands supported by TairDocs.

Overview

A TairDoc is a document data structure. You can use TairDocs to add, modify, query, or delete JSON data.

TairDoc has the following features:

- Supports JSON standards.
- Fully compatible with RedisJSON.
- Supports the syntax of JSONPath and JSON Pointer.
- Stores data in a binary tree, which simplifies the retrieval of child elements.
- Supports conversion from the JSON format to the XML or YAML Ain't Markup Language (YAML) format.

Prerequisites

The commands for TairDocs that are described in this topic can take effect only if the following conditions are met:

- Performance-enhanced instances of KVStore for Redis Enterprise Edition are used.
- The TairDoc to be managed is stored on the performance-enhanced instance.

Commands

TairDoc commands

Statement	Syntax	Description
JSON.SET	JSON.SET <key> <path> <json> [NX or XX]	Writes a JSON value to the path of a specified key. If the specified key does not exist, the path must be the root directory. If the specified key and path exist, the specified JSON value overwrites the current JSON value in the path.
JSON.GET	JSON.GET <key> [PATH] [FORMAT <XML/YAML>] [ROOTNAME <root>] [ARRNAME <arr>]	Retrieves JSON data from the TairDoc path of a specified key.
JSON.DEL	JSON.DEL <key> [path]	Deletes JSON data from the TairDoc path of a specified key. If the path is not specified, the key is deleted. If the key or path does not exist, this command does not take effect.
JSON.TYPE	JSON.TYPE <key> [path]	Retrieves the type of JSON data from the TairDoc path of a specified key.
JSON.NUMINCRBY	JSON.NUMINCRBY <key> [path] <value>	Increases JSON data in a TairDoc path by a specified value. The path must exist, and the JSON data and increased value must be of the same type, int or double.

Statement	Syntax	Description
JSON.STRAPPEND	JSON.STRAPPEND <key> [path] <json-string>	Appends a string specified in json-string to the end of the string in a TairDoc path. If you do not specify the path, the root directory is used.
JSON.STRLEN	JSON.STRLEN <key> [path]	Retrieves the JSON value length in a TairDoc path. If you do not specify the path, the root directory is used.
JSON.ARRAPPEND	JSON.ARRAPPEND <key> <path> <json> [<json> ...]	Appends one or more JSON values to the end of an array in a TairDoc path.
JSON.ARRPOP	JSON.ARRPOP <key> <path> [index]	Removes an element specified by the index parameter from an array in a specified TairDoc path and returns the removed element.
JSON.ARRINSERT	JSON.ARRINSERT <key> <path> <index> <json> [<json> ...]	Adds one or more JSON elements to an array in a TairDoc path. The index parameter specifies the position to which the JSON elements are added.
JSON.ARRLEN	JSON.ARRLEN <key> [path]	Retrieves the length of the array in a TairDoc path.
JSON.ARRTRIM	JSON.ARRTRIM <key> <path> <start> <stop>	Trims a JSON array in a specified TairDoc path. The start value and the stop value specify the range in which the JSON data is retained.
DEL	DEL <key> [key ...]	Deletes one or more TairDocs.

JSON.SET

- Syntax

JSON.SET <key> <path> <json> [NX | XX]

- Time complexity

O(N)

- Description

This command is used to write a JSON value to the path of a specified key. If the specified key does not exist, the path must be the root directory. If the specified key and path exist, the specified JSON value overwrites the current JSON value in the path.

- Parameters and options

Parameter/option	Description
key	The key of the TairDoc that you want to manage.
path	The TairDoc path in which you want to manage JSON data. - If the specified key does not exist, the path must be the root directory. - If the specified key and path exist, the specified JSON value overwrites the current JSON value in the path.

Parameter/option	Description
json	If the specified key and path exist, the specified JSON value overwrites the current JSON value in the TairDoc.
NX	Specifies that a JSON value is written only if the required path does not exist.
XX	Specifies that a JSON value is written only if the required path exists.

- Returned values
 - If the operation is successful, OK is returned.
 - null: The operation fails. This occurs when you specify the NX or XX parameter.
 - Otherwise, an exception is returned.
- Example

```
127.0.0.1:6379> JSON.SET doc . '{"foo": "bar", "baz": 42}'
OK
127.0.0.1:6379> JSON.SET doc.foo "flower"
OK
127.0.0.1:6379> JSON.GET doc.foo
"flower"
127.0.0.1:6379> JSON.SET doc.not-exists 123 XX
127.0.0.1:6379> JSON.SET doc.not-exists 123 NX
OK
127.0.0.1:6379> JSON.GET doc.not-exists
123
```

JSON.GET

- Syntax


```
JSON.GET <key> <path> [FORMAT <XML | YAML>] [ROOTNAME <root>] [ARRNAME <arr>]
```
- Time complexity

$O(N)$
- Description

This command is used to retrieve JSON data from the TairDoc path of a specified key.
- Parameters and options

Parameter/option	Description
key	The key of the TairDoc that you want to manage.
path	The TairDoc path in which you want to manage JSON data.
FORMAT	The format of the JSON data to be returned. Valid values: XML and YAML.
ROOTNAME	The tag that specifies a root element in an XML document.

Parameter/option	Description
ARRNAME	The tag that specifies an array element in an XML document.

- Returned values
 - If the operation is successful, the JSON data stored in the path is returned.
 - Otherwise, an exception is returned.
- Example

```
127.0.0.1:6379> JSON.SET doc . '{"foo": "bar", "baz": 42}'
OK
127.0.0.1:6379> JSON.GET doc
{"foo": "bar", "baz": 42}
127.0.0.1:6379> JSON.GET doc .foo
"bar"
127.0.0.1:6379> JSON.GET doc .not-exists
ERR pointer illegal or array index error or object type is not array or map
127.0.0.1:6379> JSON.GET doc . format xml
<?xml version="1.0" encoding="UTF-8"?><root><foo>bar</foo><baz>42</baz></root>
127.0.0.1:6379> JSON.GET doc . format xml rootname ROOT arrname ARRAY
<?xml version="1.0" encoding="UTF-8"?><ROOT><foo>bar</foo><baz>42</baz></ROOT>
127.0.0.1:6379> JSON.GET doc . format yaml
foo: bar
baz: 42
```

JSON.DEL

- Syntax

JSON.DEL <key> [path]
- Time complexity

$O(N)$
- Description

This command is used to delete JSON data from the TairDoc path of a specified key. If the path is not specified, the key is deleted. If the key or path does not exist, this command does not take effect.
- Parameters and options

Parameter/option	Description
key	The key of the TairDoc that you want to manage.
path	The TairDoc path in which you want to manage JSON data.

- Returned values
 - If the operation is successful, a value of 1 is returned.
 - A value of 0 is returned if the deletion fails.

- Otherwise, an exception is returned.
- Example

```
127.0.0.1:6379> JSON.SET doc '{"foo": "bar", "baz": 42}'
OK
127.0.0.1:6379> JSON.DEL doc.foo
1
127.0.0.1:6379> JSON.DEL doc.not-exists
ERR old item is null for remove or replace
127.0.0.1:6379> JSON.DEL not-exists
0
127.0.0.1:6379> JSON.GET doc
{"baz":42}
127.0.0.1:6379> JSON.DEL doc
1
127.0.0.1:6379> JSON.GET doc
127.0.0.1:6379>
```

JSON.TYPE

- Syntax
JSON.TYPE <key> [path]
- Time complexity
 $O(N)$
- Description
This command is used to retrieve the type of JSON data from the TairDoc path of a specified key.
- Parameters and options

Parameter/option	Description
key	The key of the TairDoc that you want to manage.
path	The TairDoc path in which you want to manage JSON data.

- Returned values
 - If the operation is successful, the type of JSON data is returned. The types include boolean, null, number, string, array, object, raw, reference, and const.
 - null: the specified key or path does not exist.
 - Otherwise, an exception is returned.
- Example

```
127.0.0.1:6379> JSON.SET doc . '{"foo": "bar", "baz": 42}'
OK
127.0.0.1:6379> JSON.TYPE doc
object
127.0.0.1:6379> JSON.TYPE doc.foo
string
127.0.0.1:6379> JSON.TYPE doc.baz
number
127.0.0.1:6379> JSON.TYPE doc.not-exists
127.0.0.1:6379>
```

JSON.NUMINCRBY

- Syntax

JSON.NUMINCRBY <key> [path] <value>

- Time complexity

O(N)

- Description

This command is used to increase JSON data in a TairDoc path by a specified value. The path must exist, and the JSON data and increased value must be of the same type, int or double.

- Parameters and options

Parameter/option	Description
key	The key of the TairDoc that you want to manage.
path	The TairDoc path in which you want to manage JSON data.
value	The increment to be added to the JSON data in the specified path.

- Returned values

- If the operation is successful, the increased value in the specified path is returned.
- Otherwise, an exception is returned.

- Example

```
127.0.0.1:6379> JSON.SET doc . '{"foo": "bar", "baz": 42}'
OK
127.0.0.1:6379> JSON.NUMINCRBY doc.baz 1
43
127.0.0.1:6379> JSON.NUMINCRBY doc.baz 1.5
44.5
127.0.0.1:6379> JSON.NUMINCRBY doc.foo 1
ERR node not exists or not number type
127.0.0.1:6379> JSON.NUMINCRBY doc.not-exists 1
ERR node not exists or not number type
127.0.0.1:6379>
```

JSON.STRAPPEND

- Syntax

JSON.STRAPPEND <key> [path] <json-string>

- Time complexity

$O(N)$

- Description

This command is used to append a string specified in json-string to the end of the string in a TairDoc path. If you do not specify the path, the root directory is used.

- Parameters and options

Parameter/option	Description
key	The key of the TairDoc that you want to manage.
path	The TairDoc path in which you want to manage JSON data.
json-string	The string to be appended to the specified path.

- Returned values

- If the operation is successful, the length of the increased value in the path is returned.
- If the specified key does not exist, a value of -1 is returned.
- Otherwise, an exception is returned.

- Example

```
127.0.0.1:6379> JSON.SET doc . '{"foo": "bar", "baz": 42}'
OK
127.0.0.1:6379> JSON.STRAPPEND doc .foo rrrrr
8
127.0.0.1:6379> JSON.GET doc .foo
"barrrrrr"
127.0.0.1:6379> JSON.STRAPPEND doc .not-exists
ERR node not exists or not string type
127.0.0.1:6379> JSON.STRAPPEND not-exists abc
-1
```

JSON.STRLEN

- Syntax

JSON.STRLEN <key> [path]

- Time complexity

$O(N)$

- Description

This command is used to retrieve the JSON value length in a TairDoc path. If you do not specify the path, the root directory is used.

- Parameters and options

Parameter/option	Description
key	The key of the TairDoc that you want to manage.
path	The TairDoc path in which you want to manage JSON data.

- Returned values

- If the operation is successful, the length of the value in the path is returned.
- If the specified key does not exist, a value of -1 is returned.
- Otherwise, an exception is returned.

- Example

```
127.0.0.1:6379> JSON.SET doc . '{"foo": "bar", "baz": 42}'
OK
127.0.0.1:6379> JSON.STRLEN doc .foo
3
127.0.0.1:6379> JSON.STRLEN doc .baz
ERR node not exists or not string type
127.0.0.1:6379> JSON.STRLEN not-exists
-1
```

JSON.ARRAPPEND

- Syntax

JSON.ARRAPPEND <key> <path> <json> [<json> ...]

- Time complexity

O(M×N). M specifies the number of JSON elements to be appended and N specifies the number of elements in the array.

- Description

This command is used to append one or more JSON values to the end of an array in a TairDoc path.

- Parameters and options

Parameter/option	Description
key	The key of the TairDoc that you want to manage.
path	The TairDoc path in which you want to manage JSON data.
json	The JSON value to be inserted into a specified array.

- Returned values

- If the operation is successful, the number of elements in the array is returned. The added elements are included.
- If the specified key does not exist, a value of -1 is returned.

- Otherwise, an exception is returned.
- Example

```
127.0.0.1:6379> JSON.SET doc . '{"id": [1,2,3]}'
OK
127.0.0.1:6379> JSON.GET doc .id
[1,2,3]
127.0.0.1:6379> JSON.ARRAPPEND doc .id null false true
6
127.0.0.1:6379> JSON.GET doc .id
[1,2,3,null,false,true]
127.0.0.1:6379> JSON.GET doc .id.2
3
127.0.0.1:6379> JSON.ARRAPPEND not-exists .a 1
-1
```

JSON.ARRPOP

- Syntax
JSON.ARRPOP <key> <path> [index]
- Time complexity
 $O(M \times N)$. M specifies the child elements that the specified key contains and N specifies the number of elements in the array.
- Description
This command is used to remove an element specified by an index from an array in a specified TairDoc path and return the removed element.
- Parameters and options

Parameter/option	Description
key	The key of the TairDoc that you want to manage.
path	The TairDoc path in which you want to manage JSON data.
index	The index of the array, which specifies the value to be removed. If you do not configure this parameter, the last value in the array is removed. A negative value specifies reverse numbering from the end of the array.

- Returned values
 - If the operation is successful, the removed element is returned.
 - If the array is empty, the "ERR array index outflow" error message is returned.
 - Otherwise, an exception is returned.
- Example

```
127.0.0.1:6379> JSON.SET doc . '{"id": [1,2,3]}'
OK
127.0.0.1:6379> JSON.ARRPOP doc .id 1
2
127.0.0.1:6379> JSON.GET doc .id
[1,3]
127.0.0.1:6379> JSON.ARRPOP doc .id -1
3
127.0.0.1:6379> JSON.GET doc .id
[1]
127.0.0.1:6379> JSON.ARRPOP doc .id 10
ERR array index outflow
127.0.0.1:6379> JSON.ARRPOP doc .id
1
127.0.0.1:6379> JSON.ARRPOP doc .id
ERR array index outflow
127.0.0.1:6379>
```

JSON.ARRINSERT

- Syntax

JSON.ARRINSERT <key> <path> <index> <json> [<json> ...]

- Time complexity

$O(M \times N)$. M specifies the number of JSON elements to be appended and N specifies the number of elements in the array.

- Description

This command is used to add one or more JSON elements to an array in a TairDoc path. The index parameter specifies the position to which the JSON elements are added.

- Parameters and options

Parameter/option	Description
key	The key of the TairDoc that you want to manage.
path	The TairDoc path in which you want to manage JSON data.
index	The index of the array, which specifies the value to be removed. If you do not configure this parameter, the last value in the array is removed. A negative value specifies reverse numbering from the end of the array.
json	The JSON value to be inserted into a specified array.

- Returned values

- If the operation is successful, the number of elements in the array is returned. The added elements are included.
- If the array is empty, the "ERR array index outflow" error message is returned.
- Otherwise, an exception is returned.

- Example

```
127.0.0.1:6379> JSON.SET doc . '{"id": [2,3,5]}'
OK
127.0.0.1:6379> JSON.ARRINSERT doc .id 0 0 1
5
127.0.0.1:6379> JSON.GET doc .id
[0,1,2,3,5]
127.0.0.1:6379> JSON.ARRINSERT doc .id 4 4
6
127.0.0.1:6379> JSON.GET doc .id
[0,1,2,3,4,5]
127.0.0.1:6379>
```

JSON.ARRLEN

- Syntax

JSON.ARRLEN <key> [path]

- Time complexity

$O(N)$

- Description

This command is used to retrieve the length of the array in a TairDoc path.

- Parameters and options

Parameter/option	Description
key	The key of the TairDoc that you want to manage.
path	The TairDoc path in which you want to manage JSON data.

- Returned values

- If the operation is successful, the length of the queried array is returned.
- If the specified key does not exist, a value of -1 is returned.
- Otherwise, an exception is returned.

- Example

```
127.0.0.1:6379> JSON.SET doc . '{"id": [2,3,5]}'
OK
127.0.0.1:6379> JSON.ARRLEN doc .id
3
127.0.0.1:6379> JSON.ARRLEN not-exists
-1
```

JSON.ARRTRIM

- Syntax

JSON.ARRTRIM <key> <path> <start> <stop>

- Time complexity

$O(N)$

- Description

This command is used to trim a JSON array in a TairDoc path. The start value and the stop value specify the range in which the JSON data is retained.

- Parameters and options

Parameter/option	Description
key	The key of the TairDoc that you want to manage.
path	The TairDoc path in which you want to manage JSON data.
start	The start of the range in which elements are retained after a trim. The value is an index that starts from 0. The element at the start position is retained.
stop	The end of the range in which elements are retained after a trim. The value is an index that starts from 0. The element at the end position is retained.

- Returned values

- If the operation is successful, the length of the trimmed array is returned.
- If the specified key does not exist, a value of -1 is returned.
- Otherwise, an exception is returned.

- Example

```
127.0.0.1:6379> JSON.SET doc . '{"id": [1,2,3,4,5,6]}'
OK
127.0.0.1:6379> JSON.ARRTRIM doc .id 3 4
2
127.0.0.1:6379> JSON.GET doc .id
[4,5]
127.0.0.1:6379> JSON.ARRTRIM doc .id 3 4
ERR array index outflow
127.0.0.1:6379> JSON.ARRTRIM doc .id -2 -5
ERR array index outflow
127.0.0.1:6379>
```

JSON Pointer and JSONPath

TairDoc supports the JSONPointer syntax and also supports some of the JSONPath syntax. The following table shows the syntax examples.

JSONPointer	JSONPath
<pre>127.0.0.1:6379> JSON.SET doc . '{"foo": "bar", "baz": [1,2,3]}' OK 127.0.0.1:6379> JSON.GET doc .foo "bar" 127.0.0.1:6379> JSON.GET doc .baz[0] 1</pre>	<pre>127.0.0.1:6379> JSON.SET doc "" '{"foo": "bar", "baz": [1,2,3]}' OK 127.0.0.1:6379> JSON.GET doc /foo "bar" 127.0.0.1:6379> JSON.GET doc /baz/0 1</pre>

The following table shows how TairDoc supports JSONPath and JSON Pointer.

Item	JSONPath	JSONPointer
Root element	.	""
An individual element in a path	.a.b.c	/a/b/c
Array	.a[2]	/a/2
Multiple elements in a path	.a["b.c"]	/a/b.c
	.a['b.c']	/a/b.c

11.3. Benefits

High performance

- Supports cluster instances with a memory capacity of 128 GB or larger. The instances can meet large capacity and high performance requirements.
- Supports master-replica instances with a maximum memory capacity of 32 GB. The instances can meet general capacity and performance requirements.
- Supports CPUs, disks, memory, and network interface controllers (NICs) of different specifications in a cluster without affecting the operational performance of the cluster. This ensures compatibility with your existing devices.

Elastic scaling

- Easy scaling: You can scale the instance storage capacity with only a few clicks by using the console.
- Online scaling: You can scale the instance storage capacity without service interruption.

Resource isolation

- Supports instance-level resource isolation among different instances. This ensures the stability of individual services.
- Supports multi-tenant isolation to ensure that each instance can use exclusive resources, such as CPU, memory, I/O resources, and disks.
- Supports multi-tenant parallel execution on a cluster by using multiple instances. Tasks from tenants are submitted to queues on different instances for execution. KVStore for Redis isolates resources

among tenants based on different instances.

High data security

- Data persistence: KVStore for Redis provides high-speed data read/write capabilities and enables data persistence by using a hybrid storage of memory and disks. KVStore for Redis allows you to load data from a persistent database into a cache database.
- Master/replica backup: KVStore for Redis maintains two backup copies of all data on a master node and a replica node to prevent data loss.
- Access control: KVStore for Redis supports password authentication to ensure secure and reliable access to databases.
- Data transmission encryption: KVStore for Redis supports encryption based on SSL and Secure Transport Layer (TLS) to secure data transmission.

High availability

- Master-replica architecture: Each instance runs in a master-replica architecture to eliminate the risk of single points of failure (SPOFs) and ensure high availability.
- Automatic failure detection and recovery: The system automatically detects hardware failures and performs a failover within a few seconds after a failure occurs. This minimizes the adverse impact caused by unexpected hardware failures.
- Supports automatic fault tolerance for server disk failures in a cluster, and supports hot swapping of disks. If a disk fails, services can be recovered within two minutes.

Easy-to-use

- KVStore for Redis is compatible with Redis commands. You can use a Redis client to connect to a KVStore for Redis instance and manage data.
- Supports multiple commands in each query.

Permission management

- Supports data access permission management, such as the logon permissions, table creation permissions, read and write permissions, and whitelist control permissions.
- Allows you to log on to the KVStore for Redis console to manage permissions on access control, including administrative rights settings.
- KVStore for Redis provides a unified permission management feature. This feature allows you to manage various permissions for each component of the system in the KVStore for Redis console. This isolates common users from internal permission management details, simplifies the permission management for administrators, and improves the user experience of permission management.
- Allows you to manage multiple tenants in a centralized manner in the console. For example, you can dynamically configure and manage tenant resources, isolate resources, view statistics on resource usage, and manage tenants at multiple levels.

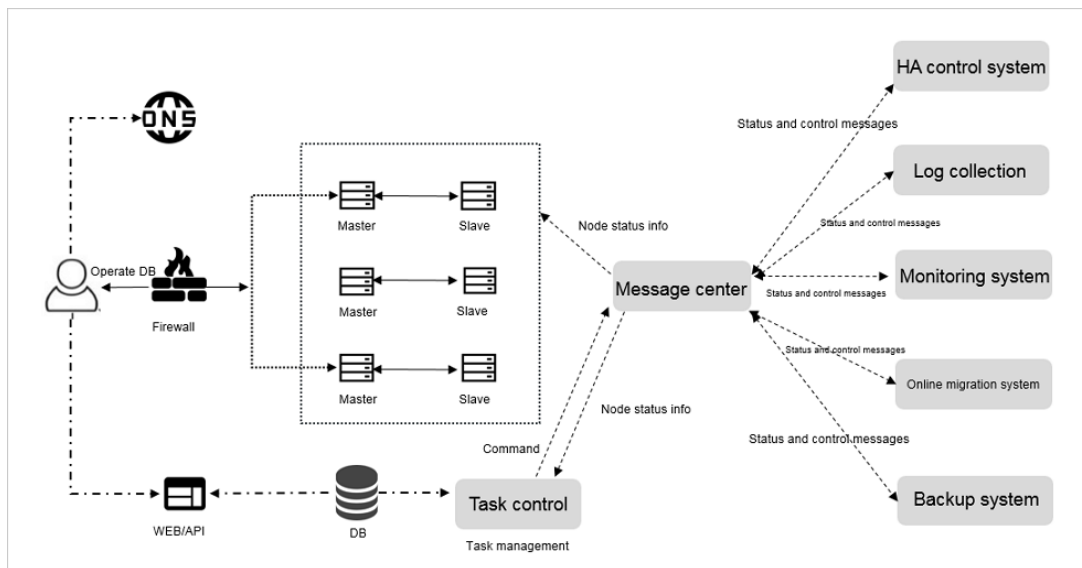
Scheduling

Supports multi-cluster scheduling, multi-resource pool scheduling, and multi-tenant scheduling.

11.4. Architecture

This topic describes the basic architecture and components of KVStore for Redis.

Architecture diagram



KVStore for Redis automatically builds a master-replica architecture.

- HA control system

A high-availability (HA) detection module is used to detect and monitor the status of KVStore for Redis instances. If this module detects that a master node is unavailable, the module automatically performs a failover to ensure high availability of KVStore for Redis instances.

- Log collection module

This module collects various logs, such as slow query logs and access control logs.

- Monitoring system module

This module collects the performance monitoring information of KVStore for Redis instances, which includes basic group monitoring, key group monitoring, and string group monitoring.

- Online migration system module

If an error occurs on the physical server that hosts a KVStore for Redis instance, this module recreates an instance based on the backup files that are stored in the backup system. This ensures high availability of your workloads.

- Backup system module

This module creates backups from KVStore for Redis instances and stores the backup files in Object Storage Service (OSS) buckets. This module allows you to retain backup files for up to seven days and customize backup settings.

- Task control module

KVStore for Redis instances support various management and control tasks, such as instance creation, configuration change, and instance backup. The task system controls and tracks tasks and manages errors based on your requirements.

11.5. Features

- KVStore for Redis Enhanced Edition adopts the high-availability technology to ensure the stability of services.

The system synchronizes data between the master node and replica node in real time. If the master node fails, the system automatically fails over to the replica node and restores services within a few seconds to ensure the high availability of the services. The replica node takes over the role of the master node. During the failover process, you can manage your workloads without service interruption.

A cluster instance runs in a distributed architecture. Each node of the instance uses a high-availability master-replica structure to automatically perform failovers and disaster recovery. This mechanism ensures the high availability of the KVStore for Redis service.

- KVStore for Redis Enhanced Edition provides a backup and recovery feature and supports custom backup policies.

You can manually back up data at any time in the console. You can also customize automatic backup policies. The system retains backup files for up to seven days. You can use the backup files to restore data with a few clicks to minimize data loss caused by user errors.

- KVStore for Redis Enhanced Edition implements multiple network security measures to secure your data.

The security features of KVStore for Redis Enhanced Edition include a virtual private cloud (VPC) network that isolates network transmission at the transport layer and an anti-DDoS service that monitors and mitigates DDoS attacks. This version also allows a whitelist to contain up to 1,000 IP addresses or CIDR blocks to prevent malicious login attempts.

- KVStore for Redis Enhanced Edition uses kernel optimization to prevent vulnerability attacks.

Alibaba Cloud performed an in-depth engine optimization on the Redis source code to prevent memory shortage, fix security vulnerabilities, and protect your business.

- KVStore for Redis Enhanced Edition supports elastic scaling to eliminate capacity and performance bottlenecks.

KVStore for Redis allows you to increase the memory capacity of your instance based on your business requirements.

The cluster architecture enables the elastic scaling of storage space and throughput performance of the database service, which eliminates performance bottlenecks.

- KVStore for Redis Enhanced Edition has multiple instances whose configurations can be changed to meet business requirements.

You can modify the configurations of the instances with standalone cache architecture and master-replica storage architecture based on your business requirements.

- KVStore for Redis Enhanced Edition provides a monitoring feature that allows you to check the status of an instance in real time.

You can check the status of an instance in real time based on the metrics that KVStore for Redis collects. These metrics include CPU utilization, number of concurrent connections, and disk usage.

- KVStore for Redis Enhanced Edition has a visualized management platform that simplifies operations and maintenance.

The KVStore for Redis console is a visualized management platform in which you can perform frequent and risky operations, such as instance cloning, backup, and data recovery.

- KVStore for Redis Enhanced Edition automatically upgrades the engine version to fix software flaws.

KVStore for Redis instances automatically upgrade engine versions to fix flaws at the earliest opportunity and keep your system updated. This way, you can easily manage the versions of your engines.

- KVStore for Redis Enhanced Edition supports custom parameter configurations.

You can configure the parameters in the KVStore for Redis console to utilize system resources.

- KVStore for Redis Enhanced Edition supports asynchronous replication.

The global replica instance synchronizes data asynchronously among child instances to minimize the impact on the service performance.

- A redundancy design is adopted for each system component to eliminate the risk of single points of failure.

11.6. Scenarios

Game industry applications

KVStore for Redis can serve as an important architecture component in the gaming industry.

Scenario 1: Use KVStore for Redis as a storage database

Gaming applications can be deployed in a simple architecture, in which the main program runs on an Elastic Compute Service (ECS) instance and the business data is stored in KVStore for Redis. KVStore for Redis can be used for persistent storage. It uses a master-replica architecture to implement redundancy.

Scenario 2: Use KVStore for Redis as a cache to accelerate connections to applications

You can use KVStore for Redis as a cache to accelerate connections to applications. You can store data in a Relational Database Service (RDS) database that is used as a backend database.

The high availability of KVStore for Redis is essential to your business. If your KVStore for Redis service becomes unavailable, the RDS instances may be overwhelmed by the requests that are sent from your applications. KVStore for Redis adopts the master-replica architecture to ensure high availability. In this architecture, the primary node provides services for your business. If this node fails, the system automatically switches workloads to the secondary node. The complete failover process is transparent.

Live streaming applications

Live streaming service can use KVStore for Redis to store user data and relationship information.

High availability

KVStore for Redis can be deployed in a master-replica architecture to significantly improve service availability.

High performance

KVStore for Redis provides cluster instances to eliminate the performance bottleneck caused by the Redis single-thread mechanism. Cluster instances can effectively handle traffic bursts during live streaming and support high performance.

High scalability

KVStore for Redis allows you to deal with traffic spikes during peak hours by scaling out an instance with a few clicks. The upgrade is completely transparent to users.

E-commerce industry applications

In the e-commerce industry, the KVStore for Redis service is widely used in modules such as commodity presentation and recommendations.

Scenario 1: online shopping systems

An online shopping system is overwhelmed by user traffic during large promotional activities such as flash sales. Most databases cannot handle the heavy load.

To resolve this issue, you can use KVStore for Redis for persistent storage.

Scenario 2: inventory management systems that support stocktaking

KVStore for Redis can be used to count the inventory and RDS can be used to store information about the quantities of items. This way, the KVStore for Redis instance reads count data and the RDS database stores count data. KVStore for Redis is deployed on a physical server. The system provides a high-level data storage capacity based on solid-state drive (SSD) storage that has high performance.

11.7. Limits

Item	Description
List data type	The number of lists is unlimited. The size of each element in the list must be 512 MB or less. We recommend that you set the number of elements in a list to a value less than 8,192. The value length is 1 MB or less.
Set data type	The number of sets is unlimited. The size of each element is 512 MB or less. We recommend that the number of elements in a set is less than 8,192. The value length is 1 MB or less.
Sorted set data type	The number of sorted sets is unlimited. The size of each element is 512 MB or less. We recommend that the number of elements in a sorted set is less than 8,192. The value length is 1 MB or less.
Hash data type	The number of fields is unlimited. The size of each element in a hash table is 512 MB or less. We recommend that the number of elements in a hash table is less than 8,192. The value length is 1 MB or less.
Number of databases	Each instance supports 256 databases.
Policies that are applied to delete expired data	• Active expiration: the system periodically detects and deletes expired keys in the background. • Passive expiration: the system deletes expired keys when you access these keys.
Idle connection recycling mechanism	KVStore for Redis does not automatically recycle idle connections to KVStore for Redis instances. You must manually recycle the connections.
Data persistence policy	KVStore for Redis uses the AOF_FSYNC_EVERYSEC method and runs the <code>fsync</code> command at an interval of 1 second.

11.8. Terms

Term	Description
Redis	KVStore for Redis is a high-performance, key-value storage system that supports caching and storage. The system is developed based on BSD open source protocols.
Redis Instance ID	Each instance corresponds to a user space, and serves as the basic unit of the KVStore for Redis service. KVStore for Redis has limits on instance specifications, such as connections, bandwidth, and CPU capacity. The limits vary based on the instance type. In the KVStore for Redis console, you can view the IDs of the instances that you purchased.
Master-replica instance	A KVStore for Redis instance that runs a master-replica architecture. A standard master-replica instance provides limited capacity and performance. However, you can change a master-replica instance to a cluster instance.
Cluster instance	A KVStore for Redis instance that runs in a cluster architecture. Cluster instances provide higher scalability and performance. However, the instances also provide limited features.
Endpoint	The endpoint that is used to connect to a KVStore for Redis instance. The endpoint is displayed as a domain name. To obtain the endpoint, choose Instance Information > Connection Information .
Eviction policy	The eviction policy is consistent with the eviction policy of open source Redis. For more information, see Eviction policy of Redis .
DB	Database Each KVStore for Redis instance supports 256 databases: DB 0 to DB 255. By default, data is written to DB 0.

11.9. Instance specifications

KVStore for Redis provides instances of Community Edition and performance-enhanced instances of Enhanced Edition. These instance types support the standard architecture and the cluster architecture. This topic describes the specifications of different instance types, such as memory capacity, maximum number of connections, and maximum bandwidth value.

Calculation rules for bandwidth values

- If network resources are sufficient, bandwidth is unlimited. However, if network resources are insufficient, the bandwidth is limited.
- The upper limit of the total bandwidth for a read/write splitting instance is 2,048 MB/s. After the upper limit is reached, the bandwidth cannot be increased even if you add more shards or nodes.
- The bandwidth value applies to the upstream and downstream bandwidths. For example, if the bandwidth of an instance is 10 MB/s, the upstream and downstream bandwidths of the instance are both 10 MB/s.
- The bandwidth in the preceding table is the internal bandwidth of the ApsaraDB for Redis instance. The Internet bandwidth is determined by the internal bandwidth and is limited by the bandwidth of the connection between the instance and a client. We recommend that you use an internal network to maximize the performance of the instance.

Performance-enhanced instances of KVStore for Redis Enhanced Edition (Tair)

Standard master-replica instances

Specification	Specification code	Number of I/O threads	Number of connections per second	Maximum number of concurrent connections	Total maximum bandwidth value (MB/s)
1 GB master-replica performance-enhanced instance	redis.amber.master.small.multithread	6	10000	30000	96
2 GB master-replica performance-enhanced instance	redis.amber.master.mid.multithread	6	10000	30000	96
4 GB master-replica performance-enhanced instance	redis.amber.master.standard.multithread	6	10000	30000	96
8 GB master-replica performance-enhanced instance	redis.amber.master.large.multithread	6	10000	30000	96
16 GB master-replica performance-enhanced instance	redis.amber.master.2xlarge.multithread	6	10000	30000	96
32 GB master-replica performance-enhanced instance	redis.amber.master.4xlarge.multithread	6	10000	30000	96
64 GB master-replica performance-enhanced instance	redis.amber.master.8xlarge.multithread	6	10000	30000	96

Cluster instances

Specification	Specification code	Number of I/O threads	Number of shards	Number of proxy servers	Number of connections per second	Maximum number of concurrent connections	Maximum bandwidth value per shard	Total maximum bandwidth value (MB/s)
2 GB cluster performance-enhanced instance (2 shards)	redis.amber.logic.sharding.1g.2db.0rodb.6proxy.multithread	6	2	4	50000	60000	96	192
4 GB cluster performance-enhanced instance (2 shards)	redis.amber.logic.sharding.2g.2db.0rodb.6proxy.multithread	6	2	4	50000	50000	96	192
8 GB cluster performance-enhanced instance (4 shards)	redis.amber.logic.sharding.2g.4db.0rodb.12proxy.multithread	6	4	4	50000	120000	96	384
16 GB cluster performance-enhanced instance (8 shards)	redis.amber.logic.sharding.2g.8db.0rodb.24proxy.multithread	6	8	8	50000	240000	96	768
32 GB cluster performance-enhanced instance (8 shards)	redis.amber.logic.sharding.4g.8db.0rodb.24proxy.multithread	6	8	8	50000	240000	96	768
64 GB cluster performance-enhanced instance (8 shards)	redis.amber.logic.sharding.8g.8db.0rodb.24proxy.multithread	6	8	8	50000	240000	96	768
32 GB cluster performance-enhanced instance (16 shards)	redis.amber.logic.sharding.2g.16db.0rodb.48proxy.multithread	6	16	16	50000	480000	96	1536
64 GB cluster performance-enhanced (16 shards)	redis.amber.logic.sharding.4g.16db.0rodb.48proxy.multithread	6	16	16	50000	480000	96	1536

Specification	Specification code	Number of I/O threads	Number of shards	Number of proxy servers	Number of connections per second	Maximum number of concurrent connections	Maximum bandwidth value per shard	Total maximum bandwidth value (MB/s)
128 GB cluster performance-enhanced instance (16 shards)	redis.amber.logic.sharding.8g.16db.0rodb.48proxy.multithread	6	16	16	50000	480000	96	1536
256 GB cluster performance-enhanced instance (16 shards)	redis.amber.logic.sharding.16g.16db.0rodb.48proxy.multithread	6	16	16	50000	480000	96	1536
512 GB cluster performance-enhanced instance (32 shards)	redis.amber.logic.sharding.16g.32db.0rodb.96proxy.multithread	6	32	32	50000	500000	96	2048
1,024 GB cluster performance-enhanced instance (64 shards)	redis.amber.logic.sharding.16g.64db.0rodb.192proxy.multithread	6	64	64	50000	500000	96	2048
2,048 GB cluster performance-enhanced instance (128 shards)	redis.amber.logic.sharding.16g.128db.0rodb.384proxy.multithread	6	128	128	50000	500000	96	2048
4,096 GB cluster performance-enhanced instance (256 shards)	redis.amber.logic.sharding.16g.256db.0rodb.768proxy.multithread	6	256	256	50000	500000	96	2048
8,192 GB cluster performance-enhanced instance (256 shards)	redis.amber.logic.sharding.32g.256db.0rodb.768proxy.multithread	6	256	256	50000	500000	96	2048

Community Edition instances

Standard master-replica instances

Specification	Service code	Maximum number of connections	Maximum bandwidth value (MB/s)	CPU
1 GB master-replica instance	redis.master.small.default	20000	48	1 core
2 GB master-replica instance	redis.master.mid.default	20000	48	1 core
4 GB master-replica instance	redis.master.stand.default	20000	48	1 core
8 GB master-replica instance	redis.master.large.default	20000	48	1 core
16 GB master-replica instance	redis.master.2xlarge.default	20000	48	1 core
32 GB master-replica instance	redis.master.4xlarge.default	20000	48	1 core
64 GB master-replica instance	redis.master.8xlarge.default	20000	96	1 core

Cluster instances

Specification	Service code	Number of shards	Maximum number of connections	Maximum bandwidth value (MB/s)	CPU
16 GB cluster instance	redis.logic.sharding.2g.8db.0rodb.8proxy.default	8	80000	768	8 cores
32 GB cluster instance	redis.logic.sharding.4g.8db.0rodb.8proxy.default	8	80000	768	8 cores
64 GB cluster instance	redis.logic.sharding.8g.8db.0rodb.8proxy.default	8	80000	768	8 cores
128 GB cluster instance	redis.logic.sharding.8g.16db.0rodb.16proxy.default	16	160000	1536	16 cores
256 GB cluster instance	redis.logic.sharding.16g.16db.0rodb.16proxy.default	16	160000	1536	16 cores
512 GB cluster instance	redis.logic.sharding.16g.32db.0rodb.32proxy.default	32	320000	2048	32 cores
1 TB cluster instance	redis.logic.sharding.16g.64db.0rodb.64proxy.default	64	500000	2048	64 cores

Specification	Service code	Number of shards	Maximum number of connections	Maximum bandwidth value (MB/s)	CPU
2 TB cluster instance	redis.logic.sharding.16g.128db.0rodb.128proxy.default	128	500000	2048	128 cores
4 TB cluster instance	redis.logic.sharding.16g.256db.0rodb.256proxy.default	256	500000	2048	256 cores

12. ApsaraDB for MongoDB

12.1. What is ApsaraDB for MongoDB?

ApsaraDB for MongoDB is a high-performance distributed data storage service. It is a stable, reliable, and resizable database service fully compatible with MongoDB protocols. ApsaraDB for MongoDB offers a full range of database solutions, such as disaster recovery, backup, restoration, monitoring, and alerts.

ApsaraDB for MongoDB supports the following features:

- Incorporates advanced functions such as disaster recovery failover and downtime migration.
- Supports quick database backup and restoration. Allows you to perform standard database backup and database rollback operations in the ApsaraDB for MongoDB console.
- Provides over 20 performance monitoring metrics and alerting. This helps you learn about the performance status of your database.
- Provides visual data management tools for convenient O&M.
- Enables you to create instances on VPCs.

12.2. Benefits

- High availability
 - Three-node replica sets and sharded clusters featured by high service availability

ApsaraDB for MongoDB uses high-availability architectures of three-node replica sets and sharded clusters. The data nodes are located on different physical servers and can automatically synchronize data.
 - Automatic backup and quick data restoration

Data is automatically backed up and uploaded to Object Storage Service (OSS) on a daily basis. This improves data disaster recovery capabilities and reduces disk space consumption. Backup files can be used to restore instance data to their original instance. This prevents irreversible effects on service data caused by incorrect operations and other errors.
- Data backup: ApsaraDB for MongoDB allows you to perform full or incremental backup and restore data from storage. Clusters can be backed up between data centers. The backup process is visually presented.
- High security
 - Anti-DDoS protection: ApsaraDB for MongoDB filters inbound traffic. When DDoS attacks are identified, the source IP addresses will be scrubbed. If scrubbing fails, blackhole filtering is triggered.
 - IP whitelist configuration: You can configure up to 1,000 IP addresses to connect to an ApsaraDB for MongoDB instance. The IP whitelist can directly control risks at the source.
- Audit log management: You can store audit logs in Log Service and download logs from Log Service. This facilitates long-term storage and management of audit logs.
- Ease of use: ApsaraDB for MongoDB provides various performance monitoring features. ApsaraDB for MongoDB provides real-time monitoring information about the CPU utilization, connections, and disk usage of your instances and sends alerts to keep you informed of the status of your instances.
- Scalability: Both replica set and sharded cluster instances of ApsaraDB for MongoDB support elastic

scaling. You can change the configuration of your instance if the current configuration cannot meet performance requirements or is unsuitable for your business needs. The configuration change process is completely transparent and does not affect your business.

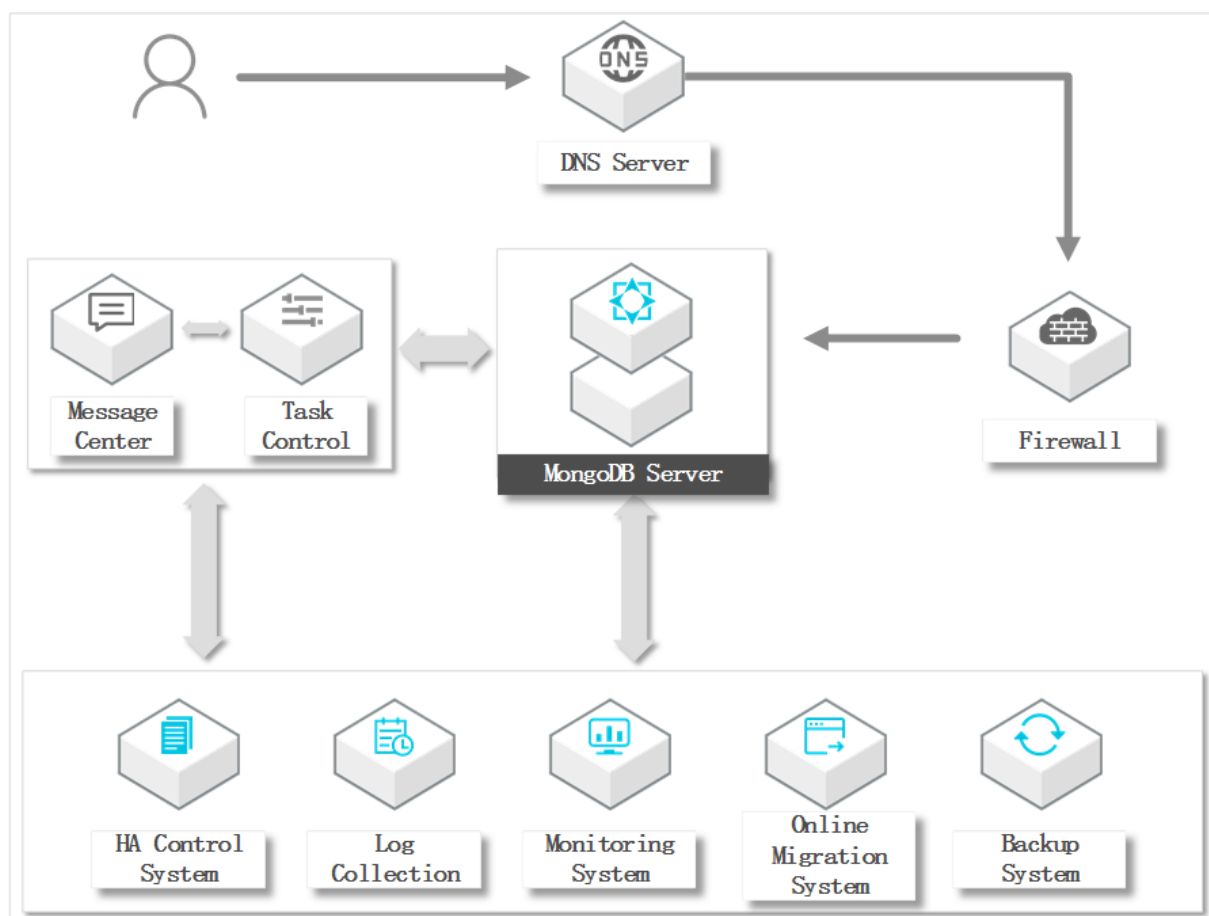
- **Isolation:** ApsaraDB for MongoDB uses multiple instances and implements tasks for multiple tenants within a cluster at the same time. The tasks of the tenants are implemented in different instances. ApsaraDB for MongoDB instances are separated to isolate resources between different tenants.
- **Permission management:** ApsaraDB for MongoDB allows you to configure and manage tenants in a dynamic and centralized manner. You can also isolate resources and query usage statistics of resources. Management of multi-level tenants is supported.
- **Scheduling:** ApsaraDB for MongoDB schedules resources from multiple resource pools for tenants within different clusters.

12.3. System architecture

12.3.1. ApsaraDB for MongoDB system architecture

This topic describes the architectures and components of ApsaraDB for MongoDB.

The following figure shows the ApsaraDB for MongoDB system architecture.

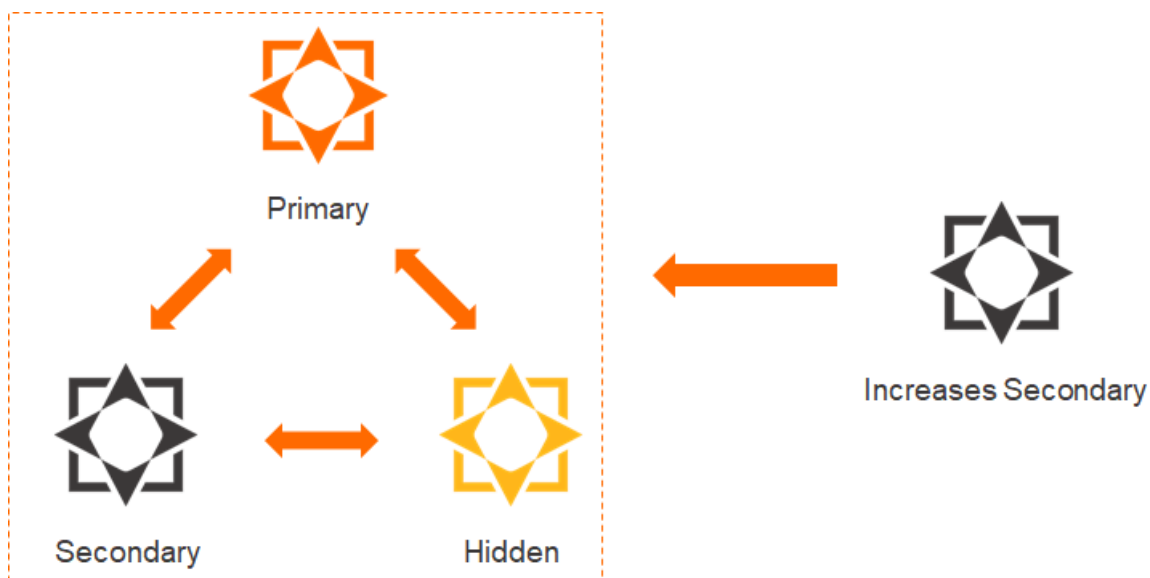


- **HA control system:** This module checks the high availability of an instance. You can use this module to detect and monitor the running status of ApsaraDB for MongoDB instances. If the system determines that the primary node of an ApsaraDB for MongoDB instance is unavailable, it fails over to a secondary node to ensure the high availability of the instance.
- **Log collection:** This module collects MongoDB running logs, including the slow query log and access log of an instance.
- **Monitoring system:** This module collects performance monitoring information about MongoDB instances, including basic metrics, disk capacity, network requests, and the number of operations that you have performed.
- **Online migration system:** When an error occurs with the physical server that hosts the instances, this module recreates an instance on the fly based on the backup files stored in the backup system. This ensures the continuity of your business.
- **Backup system:** This module generates ApsaraDB for MongoDB instance backups and stores the backup files in OSS. This module allows you to configure custom backup settings and create temporary backups. Files are retained for seven days.
- **Task control:** ApsaraDB for MongoDB supports multiple management operations, such as creating instances, changing instance configurations, and backing up instances. This module controls and tracks these tasks and troubleshoots errors based on your commands.

12.3.2. Architecture of replica set instances

Replica set instances use the three-node architecture. This topic describes the three nodes of a replica set instance.

Architecture



ApsaraDB for MongoDB uses a multi-node architecture to ensure high availability. A replica set instance consists of a primary node, one or more secondary nodes, and a hidden node. You can manage primary and secondary nodes. The following section describes the three nodes of a replica set instance:

- **Primary node:** All read and write operations are performed on the primary node. Each replica set instance contains only one primary node.
- **Secondary node:** The data of a secondary node is synchronized with the primary node by using

oplogs. If the primary node fails, a secondary node can be elected as the new primary node to ensure high availability.

Note When you connect to an instance by using the endpoint of a secondary node, you can only read data from the instance but cannot write data to it.

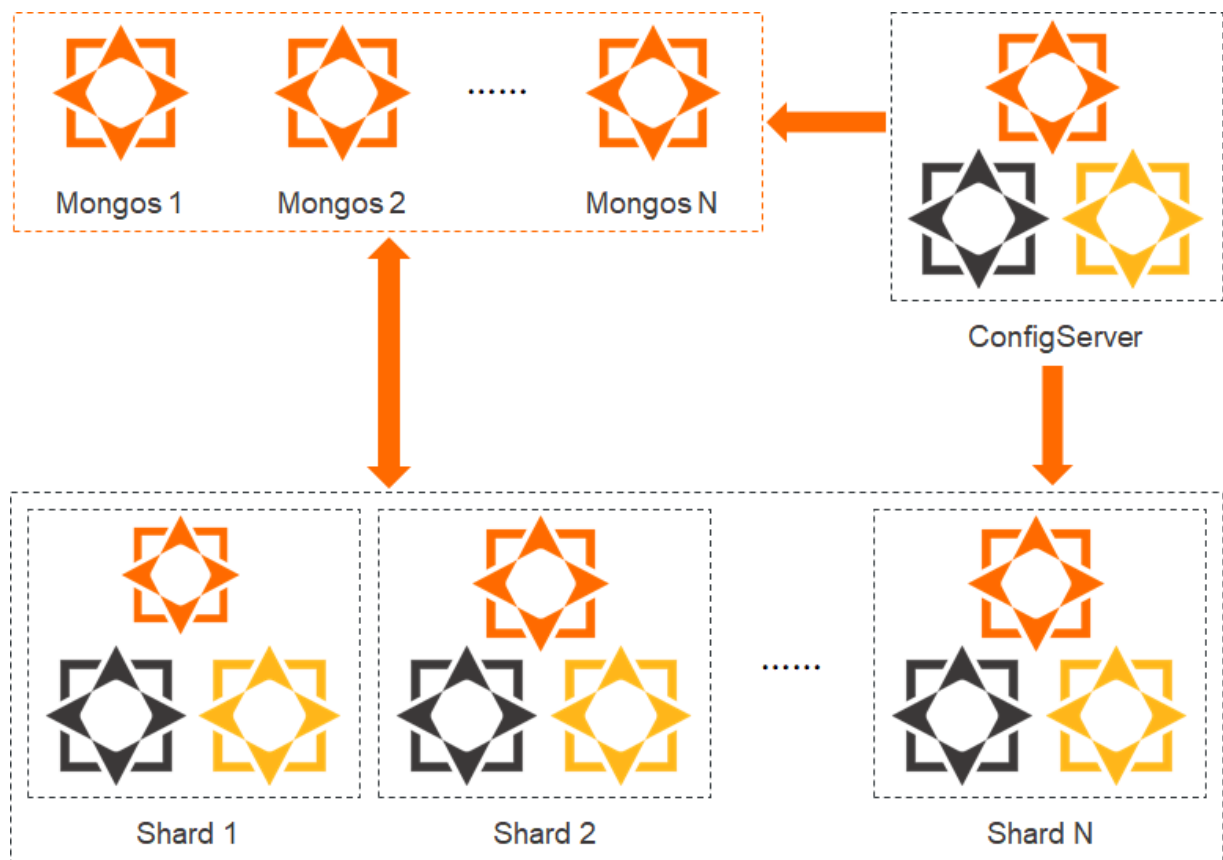
- **Hidden node:** The data of a hidden node is synchronized with the primary node by using oplogs. If a secondary node fails, the hidden node can be elected as the new secondary node to ensure high availability.

Note The hidden node is used only to ensure high availability and is not visible to users.

12.3.3. Architecture of sharded cluster instances

A sharded cluster instance consists of three components: mongos, shard, and Configserver. This topic describes the components of a sharded cluster instance.

Architecture



Components

Component	Architecture	Description
Mongos	Standalone architecture	Queries and writes routes to the corresponding shard node.

Component	Architecture	Description
Shard	Replica set architecture (three nodes)	Stores database data.
Configserver	Replica set architecture (three nodes)	Stores the metadata for clusters and shards. The metadata contains information about which data is stored in each shard.  Note You cannot change the specifications of the Configserver (1 vCPU, 2 GiB memory, and 20 GB storage capacity).

12.4. Features

Flexible architectures

ApsaraDB for MongoDB supports flexible deployment architectures such as [Architecture of replica set instances](#) and [Architecture of sharded cluster instances](#) to meet requirements of different business scenarios.

Elastic scaling

- Quick scaling of storage capacity: You can adjust the storage capacity of an instance in the ApsaraDB for MongoDB console based on business requirements.
- Storage capacity adjustment on the fly: You can adjust the storage capacity of an instance on the fly. This ensures the continuity of your business.

Data security

- Automatic backup
 - ApsaraDB for MongoDB allows you to set a time interval based on which backups are periodically created.
 - You can flexibly set the backup start time based on off-peak hours of your business.
 - Backup files are retained for free for up to seven days.
- Temporary backup
 - You can create temporary backups.
 - Backup files are retained for free for up to seven days.
- Data restoration: You can use backup files to overwrite existing data and restore instances to a previous state.
- Backup file download: ApsaraDB for MongoDB retains your backup files for free for up to seven days. During this period, you can log on to the ApsaraDB for MongoDB console and download the backup files to a local device.
- Creation of instances from backup sets: You can create an instance in the ApsaraDB for MongoDB console by using backup files. This helps you complete the deployment process with ease.
- IP whitelist configuration: ApsaraDB for MongoDB can filter IP addresses that access your instance.

You can log on to the ApsaraDB for MongoDB console and configure a whitelist of up to 1,000 IP addresses. This provides a highly secure access environment.

- Multi-layer network security protection
 - VPCs are isolated at the TCP layer.
 - Anti-DDoS can monitor and block DDoS attacks in real time.
 - You can add up to 1,000 IP addresses to the whitelist.

Intelligent O&M

- Monitoring platform

This platform provides real-time monitoring information about the CPU utilization, connections, and disk usage of your instances and sends alerts to keep you informed of the status of your instances.

- Graphical O&M platform

This platform allows you to perform frequent and high-risk operations, such as instance cloning, backup, and data restoration.

High availability of clusters

A multi-node cluster architecture is used. Each component management node in the platform implements a high availability mechanism so that the failure of a service node within a cluster does not affect the overall operation of the corresponding service instance.

12.5. Scenarios

- Scenario where read/write splitting is required

ApsaraDB for MongoDB uses a high-availability architecture that features a three-node replica set. These three data nodes are located on different physical servers. The secondary and standby nodes automatically synchronize data from the primary node. Services are provided by the primary and secondary nodes. These two nodes have separate domain names and collaborate with MongoDB drivers to distribute read requests.

- Scenario where flexibility is required

As a schema-free database, ApsaraDB for MongoDB is particularly suitable for startup businesses because it does not require you to change table schema. You can store data with fixed structures in ApsaraDB RDS databases, business data with flexible structures in ApsaraDB for MongoDB databases, and frequently accessed data in ApsaraDB for Memcache databases or KVStore for Redis databases. This helps you store data efficiently and reduce costs.

- Mobile application scenario

ApsaraDB for MongoDB supports two-dimensional space indexes. Therefore, it can provide support for location-based mobile application services. ApsaraDB for MongoDB adopts a dynamic storage method that is suitable for storing heterogeneous data from multiple systems. This satisfies the needs of mobile applications.

- IoT scenario

ApsaraDB for MongoDB provides excellent performance and an asynchronous data writing function. In special scenarios, it can provide in-memory database performance. This makes it extremely suitable for IoT writing scenarios with high concurrency. The MapReduce feature of ApsaraDB for MongoDB can aggregate and analyze large amounts of data.

- **Scenario where core log systems are required**

In scenarios where data is asynchronously written to disks, ApsaraDB for MongoDB can provide excellent data insertion performance and processing capabilities of an in-memory database. ApsaraDB for MongoDB allows you to create secondary indexes for dynamic queries. It can use the MapReduce aggregation framework to perform multidimensional data analysis.

12.6. Limits

Item	Limit
Create a replica set instance	When a replica set instance is created, three nodes are added. ApsaraDB for MongoDB provides a primary node, a secondary node, and a hidden node for each replica set instance. The hidden node is invisible to you. You cannot create secondary nodes.
Restart an instance	Instances must be restarted in the ApsaraDB for MongoDB console.

12.7. Terms

Term	Description
Region	The geographical location of the server on which the ApsaraDB for MongoDB instance runs. You must specify a region when you create an ApsaraDB for MongoDB instance. The region cannot be changed after the instance is created. When you create an ApsaraDB for MongoDB instance, you must use it with an Alibaba Cloud ECS instance. You can access ApsaraDB for MongoDB instances through internal networks. Make sure that the region of an ApsaraDB for MongoDB instance is the same as that of the corresponding ECS instance.
Instance	An ApsaraDB for MongoDB instance. An instance is the basic unit of ApsaraDB for MongoDB services that you create. An instance is the operating environment for ApsaraDB for MongoDB and exists as a separate process on a host. You can create, modify, and delete an instance in the ApsaraDB for MongoDB console. Instances are independent and their resources are isolated. An instance does not consume resources such as CPU, memory, or I/O of another instance. Each instance has its own features, such as database type and version. The system has parameters to control instance behaviors.
Memory	The maximum memory that an instance can use.
Disk capacity	The disk size that you select when you create an instance. The disk capacity occupied by the instance includes datasets and the space required for normal instance operations, such as the system database, database rollback log, redo log, and indexes. Make sure that the ApsaraDB for MongoDB instance has sufficient disk space to store data. Otherwise, the instance may be locked. If an instance is locked due to insufficient disk capacity, you can purchase a larger disk to unlock the instance.
IOPS	The maximum number of read or write operations performed on block devices per second. Each operation consumes 4 KB.

Term	Description
CPU core	The maximum computing capability of the instance. A single core CPU has a minimum of 2.3 GHz hyper-threading (Intel Xeon series Hyper-Threading) computing power.
Connections	The TCP connections between clients and ApsaraDB for MongoDB instances. If a client uses a connection pool, the connections between the client and instance are persistent connections. Otherwise, they are short connections.
Mongos	The routing service that processes requests. All requests must be coordinated through mongos that serves as a request distribution center and forwards data requests to the corresponding shard server. You can use multiple mongos to process requests. If one fails, other mongos can continue to process the requests.
Config server	The servers that store all database metadata configurations, including routers and shards. mongos does not store but caches shard server information and data routing information in memory. The information is stored on config servers. When you start mongos for the first time or shut it down and then restart it, it automatically loads configuration information from config servers. mongos updates the cache when there are metadata changes. This ensures that mongos can always obtain the correct routing information. Config servers store metadata of shards and routers and have high requirements for service availability and data reliability. Therefore, ApsaraDB for MongoDB uses a three-node replica set to ensure the reliability of the config servers.

12.8. Instance types

This topic describes the instance types available in ApsaraDB for MongoDB.

Replica set instance types

Architecture	Category	Specification	Instance type	Maximum connections	Maximum IOPS	Storage capacity
	Three-node edition	1 vCPU, 2 GiB memory	dds.mongo.mid	500	8,000	10 to 2,000 GB
		2 vCPUs, 4 GiB memory	dds.mongo.standard	1,000	8,000	
		4 vCPUs, 8 GiB memory	dds.mongo.large	3,000	8,000	
		8 vCPUs, 16 GiB memory	dds.mongo.xlarge	5,000	8,000	
		8 vCPUs, 32 GiB memory	dds.mongo.2xlarge	8,000	14,000	

Architecture	Category	Specification	Instance type	Maximum connections	Maximum IOPS	Storage capacity
Replica set instance		16 vCPUs, 64 GiB memory	dds.mongo.4xlarge	16,000	16,000	
	Dedicated CPU	2 vCPUs, 16 GiB memory	mongo.x8.medium	2,500	4,500	250 GB
		4 vCPUs, 32 GiB memory	mongo.x8.large	5,000	9,000	500 GB
		8 vCPUs, 64 GiB memory	mongo.x8.xlarge	10,000	18,000	1,000 GB
		16 vCPUs, 128 GiB memory	mongo.x8.2xlarge	20,000	36,000	2,000 GB
		32 vCPUs, 256 GiB memory	mongo.x8.4xlarge	40,000	72,000	2,000 GB
	Dedicated host	60 vCPUs, 440 GiB memory	dds.mongo.2xmonopolize	100,000	100,000	3,000 GB

Sharded cluster instance types

Node type	Category	Specification	Instance type	Maximum connections	Maximum IOPS	Storage capacity
Mongos	General purpose	1 vCPU, 2 GiB memory	dds.mongos.mid	1,000	N/A	N/A
		2 vCPUs, 4 GiB memory	dds.mongos.standard	2,000		
		4 vCPUs, 8 GiB memory	dds.mongos.large	4,000		
		8 vCPUs, 16 GiB memory	dds.mongos.xlarge	8,000		

Node type	Category	Specification	Instance type	Maximum connections	Maximum IOPS	Storage capacity
Shard		8 vCPUs, 32 GiB memory	dds.mongos.2xlarge	16,000		
		16 vCPUs, 64 GiB memory	dds.mongos.4xlarge	16,000		
	General purpose	1 vCPU, 2 GiB memory	dds.shard.mid	N/A	1,000	10 to 2,000 GB
		2 vCPUs, 4 GiB memory	dds.shard.standard		2,000	
		4 vCPUs, 8 GiB memory	dds.shard.large		4,000	
		8 vCPUs, 16 GiB memory	dds.shard.xlarge		8,000	
		8 vCPUs, 32 GiB memory	dds.shard.2xlarge		14,000	
		16 vCPUs, 64 GiB memory	dds.shard.4xlarge		16,000	
	Dedicated CPU	2 vCPUs, 16 GiB memory	dds.shard.sn8.xlarge.3		4,500	10 to 250 GB
		4 vCPUs, 32 GiB memory	dds.shard.sn8.2xlarge.3		9,000	10 to 500 GB
		8 vCPUs, 64 GiB memory	dds.shard.sn8.4xlarge.3		18,000	10 to 1,000 GB
		16 vCPUs, 128 GiB memory	dds.shard.sn8.8xlarge.3		36,000	10 to 2,000 GB
		32 vCPUs, 256 GiB memory	dds.shard.sn8.16xlarge.3		72,000	10 to 3,000 GB

Node type	Category	Specification	Instance type	Maximum connections	Maximum IOPS	Storage capacity
Configserver	General purpose	1 vCPU, 2 GiB memory	dds.cs.mid		1,000	20 GB

13.Data Management (DMS)

13.1. What is DMS?

Data Management (DMS) is a fully managed service that is provided by Alibaba Cloud. This service allows you to manage data, schema, development procedures, development specifications, users, and permissions. DMS also provides security control to ensure secure access to databases.

Supported databases

- Relational databases: MySQL, SQL Server, PostgreSQL, PolarDB-X (previously called DRDS), Oracle, and ApsaraDB for OceanBase.
- NoSQL databases: KVStore for Redis and ApsaraDB for MongoDB.
- Analytical databases: AnalyticDB for MySQL and AnalyticDB for PostgreSQL.

Features

- DMS provides support for the entire process of database development. The process includes the following stages: 1. Design table structures in an on-premises environment based on the predefined design specification. 2. Publish and produce SQL reviews that are included in code and schemas to a specified environment on demand. The preceding operations are performed before the code is released. These SQL reviews in code are used to add, remove, modify, or query rows.
- DMS provides fine-grained access control at the database, table, and field levels. You can perform all the required operations on databases in the DMS console. These operations can be traced and audited.
- DMS allows you to configure the required operation specifications and approval processes for multiple modules. These modules include the schema design, data changes, data export, and permission requests.
- DMS provides an integrated platform that connects database development with database interaction. You can manage databases without the need to use database accounts and passwords to switch between database endpoints at a high frequency.
- DMS provides the task orchestration feature. This feature allows you to orchestrate and schedule SQL tasks for databases on a regular basis. You can use this feature to perform the required operations with ease. For example, you can transfer historical data, analyze periodical reports, and generate analytical results.

13.2. Benefits

DMS provides multiple benefits. These benefits include various data sources, secure and controllable procedures, and fine-grained access control. These benefits improve data security and simplify data management.

Various data sources

- Relational databases: MySQL, SQL Server, PostgreSQL, PolarDB-X (previously called DRDS), Oracle, and ApsaraDB for OceanBase.
- NoSQL databases: ApsaraDB for Redis and ApsaraDB for MongoDB.
- Analytical databases: AnalyticDB for MySQL and AnalyticDB for PostgreSQL.

Unified operations and comprehensive audits

- After you add a database instance to DMS as an administrator, you can perform the required operations in the DMS console. These operations include querying databases, changing schemas, and changing data.
- You can query and audit all historical operations based on multiple dimensions. These dimensions include the operator, database, table, and time.

Fine-grained access control

Common users do not need to use database accounts and passwords. These users only need to request the query, export, or change permission on the destination database, table, or field in the DMS console based on their business requirements. After a permission expires, DMS revokes the permission.

Custom approval processes

You can create custom approval processes for the modules of each database instance. These approval processes are specific to your business requirements. This allows you to meet requirements from several aspects, such as efficiency and security. Example:

- Impose loose controls on a test environment. You can reduce stages or set no approval process.
- Impose strict controls on a production environment. You can specify an approval process that includes the required operations for the production environment. The production environment takes effect until all these operations are approved by the specified engineers in sequence.

Custom design specifications for schemas

You can create custom design specifications for MySQL table schemas. These design specifications include the field type, index type, number of indexes, field length, table size, and release process.

Simple procedure to schedule and orchestrate periodical tasks

DMS provides a quick method to create the required orchestration and recurring schedules for the SQL task nodes of various databases in a quick manner. You can use this feature to perform several operations on databases to explore the value of data. These operations include transferring historical data and generating periodical reports.

13.3. Architecture

DMS provides database management services in the business layer, scheduling layer, and connection layer. This architecture allows DMS to handle real-time access to databases and schedule backend data tasks.

13.4. Features

13.4.1. Workbench

Category	Description
Feature	This feature provides a user center. This user center provides multiple management shortcuts that direct you to the related pages. These shortcuts allow you to view the tickets, recent databases, followed databases, authorized databases, and owned databases.

Category	Description
Scenario	Query tickets and databases, manage user permissions, and control access to owned databases.
Procedure	Log on to the DMS console, go to the Workbench tab, and then use the required feature.
Limit	No limit is set.

13.4.2. Top search box

Category	Description
Feature	This feature allows you to query all the databases or tables of DMS.
Scenario	Query databases or tables to perform the required operations.
Procedure	Query databases or tables by name, keyword, or a combination of conditions
Limit	The control mode of the required instance must be security collaboration.

13.4.3. Permission application

Category	Description
Feature	This feature allows you to apply for the required permissions. • For example, you can request the query, export, or change permission for a database, table, field, or row. • You can request the logon permission for an instance. • You can request the ownership of an instance, database, or table.
Scenario	Grant permissions to common users.
Procedure	In the top navigation bar of the DMS console, choose Task Type > Permission Application and submit the required application.
Limit	To apply a permission to the database, table, field, or row of an instance, the control mode of the instance must be security collaboration.

13.4.4. Data Plans

13.4.4.1. Data Changes

Category	Description

Category	Description
Feature	This feature allows you perform multiple operations to change the destination database or table, for example, data definition language (DDL) and data manipulation language (DML) operations. These operations include changing common data, clearing historical data, changing data without locking tables, changing programmable objects, and exporting data in batches.
Scenario	Initialize schemas, initialize data, and rectify data during the development of a project or after the code of the project is released.
Procedure	In the top navigation bar of the DMS console, choose Data Plans > Data Changes and submit the required SQL script to be executed.
Limits	The operator must have the change permission on the destination database or table.

13.4.4.2. Change schemas without locking tables

Category	Description
Feature	This feature allows you to update the schemas of a table without the need to lock the table. This prevents business interruption or the latency issue of the secondary database due to locked tables.
Scenario	Update table schemas.
Procedure	In the top navigation bar of the DMS console, choose System > Instance, find the required instance, and then click Edit next to the instance. In the Edit instance dialog box, click Advanced information to enable the change schemas without the need to lock tables feature. Use one of the procedures to change the schemas of a table without the need to lock the table: • In the top navigation bar of the DMS console, choose Schemas > Schema Design. • In the top navigation bar of the DMS console, choose Data Plans > Data Changes > Normal Data Modify. • In the top navigation bar of the DMS console, choose System Management > Task and submit the required SQL change task.
Limit	• The operator must have the change permission on the destination database. • The control mode of the required instance must be security collaboration.

13.4.4.3. Data import

Category	Description
Feature	This feature provides a quick method for you to import a large amount of data to databases. This reduces the costs of labors and material resources.
Scenario	Import a large amount of data.

Category	Description
Procedure	In the top navigation bar of the DMS console, choose Data Plans > Data Import . Export the required contents, and submit an executable SQL script or a CSV file.
Limit	- The operator must have the change permission on the destination database. - The destination database must be one of the following databases: MySQL, ApsaraDB for OceanBase, and PolarDB-X.

13.4.4.4. Data export

Category	Description
Feature	This feature allows you to export data from the destination database or table. The data include SQL result sets, databases, and tables.
Scenario	Analyze a large amount of data.
Procedure	In the top navigation bar of the DMS console, choose Data Plans > Data Export , and export the required contents or submit an executable SQL script.
Limit	The operator must have the export permission on the destination database or table.

13.4.4.5. Test data generation

Category	Description
Feature	This feature allows you to create a custom data generation method based on the data type of a field. We recommend that you use this feature. This feature can also be used to generate a large amount of test data. This reduces the time that is required to generate test data.
Scenario	Prepare data for a test environment.
Procedure	In the top navigation bar of the DMS console, choose Data Plans > Test Data Generate and configure the required data to be generated.
Limits	- The operator must have the change permission on the destination database or table. - The destination database must be a relational database, such as MySQL.

13.4.4.6. Database cloning

Category	Description
Feature	This feature allows you to synchronize the schemas and data of a source database to a destination database.

Category	Description
Scenario	Create schemas and data for a test environment.
Procedure	In the top navigation bar of the DMS console, choose Data Plans > Database Clone and submit a ticket to clone the destination database.
Limit	• The operator must have the change permission on the destination database. • The destination database must be a relational database, such as MySQL.

13.4.5. Data factory

13.4.5.1. Task orchestration

Category	Description
Feature	This feature allows you to orchestrate SQL and DSQL nodes of various database types based on a directed acyclic graph (DAG) workflow. You can add these nodes to periodic scheduling tasks and perform the required operations. For example, you can develop data warehouses, transform periodical report, or transfer historical data. You can also use this feature to implement data integration, deploy SQL task nodes, check dependencies, and implement the periodic scheduling of sample task flows.
Scenario	Analyze periodical data and generate reports.
Procedure	In the top navigation bar of the DMS console, choose Data Factory > Task Orchestration , and configure the required task nodes to orchestrate. Then, configure periodic scheduling tasks.
Limit	• The operator must have the change permission to the destination database or table. • The control mode of the related instance must be security collaboration.

13.4.5.2. Data warehouse development

Category	Description
Feature	This feature uses databases as the computing engine and integrates a variety of tools and services in the database ecosystem. This allows you to develop and manage data warehouses with ease.
Scenario	Implement data integration, data transformation, data visualization, and data mining.
Procedure	In the top navigation bar of the DMS console, choose Data Factory > Data Warehouse Development and configure the required information. The information includes the data warehouse project and task process.

Category	Description
Limit	• The operator must assume the owner role of the destination database. • The control mode of the related instance must be security collaboration.

13.4.5.3. Data services

Category	Description
Feature	This feature allows you to define API operations based on the data that is managed in DMS. You can call the required API operations to export the managed data in a quick manner.
Scenario	Export data of minimum granularity, generate visualized data, and provide transformed data that can be read by multiple applications.
Procedure	In the top navigation bar of the DMS console, choose Data Factory > Data Service and perform the required operations on an API operation. For example, you can develop, deactivate, test, or release an API operation.
Limit	• The operator must have the query permission to the destination database or table. • The control mode of the related instance must be security collaboration.

13.4.6. Schemas

13.4.6.1. Synchronization between tables and databases

Category	Description
Feature	• This feature allows you to compare the differences between the source table and the destination table. Then, you can submit and execute differential data definition language (DDL) scripts. • This feature provides multiple tools. You can use these tools to initialize empty databases, synchronize schemas, and repair table consistency.
Scenario	Synchronize table schemas.
Procedure	In the top navigation bar of the DMS console, choose Schemas > Table Sync , and submit the required comparison task. Then, execute a diff script.
Limit	• The operator must have the query permission to the source database and the change permission to the destination database. • The source and destination databases must be MySQL or ApsaraDB for OceanBase databases.

13.4.6.2. Schema design

Category	Description
Feature	This feature allows you to design table schemas for the destination databases or tables. These table schemas conform to the predefined development specifications
Scenario	Design table schemas during the development of a project.
Procedure	In the top navigation bar of the DMS console, choose Schemas > Schema Design and edit the required table schema. Then, generate a change script and apply the changes to the destination database by executing the change script.
Limit	• The operator must have the change permission to the destination database. • The destination database must be a MySQL or ApsaraDB for OceanBase database. • The control mode of the related instance must be security collaboration.

13.4.7. SQL reviews

Category	Description
Feature	This feature allows you to review SQL statements in code.
Scenario	Review SQL statements before a project is released.
Procedure	In the top navigation bar of the DMS console, choose Optimization > SQL Review and submit the required SQL script that is used to review SQL statements.
Limit	• The operator must have the query permission on the destination database. • The destination database must be a MySQL database. • The control mode of the related instance must be security collaboration.

13.4.8. SQLConsole

Category	Description
Feature	This feature allows you to perform the required operations on the destination database, for example, query or modify the destination database. This feature provides shortcuts to the following quick actions: • Execute quick queries by opening tables. • Edit, update, or export query results. • Open a command window. • Perform the required operations on tables, for example, rename, edit, delete, modify, or optimize tables.

Category	Description
Scenario	Query or analyze data.
Procedure	Write an SQL statement for the destination database and execute the SQL statement.
Limit	The operator must have the query permission on the destination database or table.

13.4.9. Cross-instance queries

Category	Description
Feature	This feature allows you to perform real-time join queries across online heterogeneous data sources that are deployed in different environments.
Scenario	Perform quick queries and analysis on data across multiple instances.
Procedure	In the top navigation bar of the DMS console, choose System Management > Instance , and enable the cross-database query feature. Then, create database links. In the Cross-database Query dialog box, submit an executable SQL statement.
Limit	- The operator must have the query permission to the destination database or table. - The control mode of the related instance must be security collaboration.

13.4.10. System management

13.4.10.1. Instance management

Category	Description
Feature	This feature allows you to centrally manage Alibaba Cloud Apsara Stack database instances. The following databases are supported: - Relational databases: MySQL, SQL Server, PostgreSQL, PolarDB-X (previously called DRDS), Oracle, and ApsaraDB for OceanBase. - NoSQL databases: ApsaraDB for Redis and ApsaraDB for MongoDB. - Analytical databases: AnalyticDB for MySQL and AnalyticDB for PostgreSQL.
Scenario	Manage instances.
Procedure	Create a database account and password that are specific to DMS for the database instance to be managed. In the top navigation bar of the DMS console, choose System Management > Instance and add the database instance.  Note If the control mode is security collaboration, you must associate the required security rule. However, if the control mode is flexible management, you cannot associate security rules.

Category	Description
Limit	The operator must assume the administrator or DBA role.

13.4.10.2. User management

Category	Description
Feature	This feature allows you to centrally manage users in DMS. You can use this feature to manage Apsara Stack tenant accounts, RAM users, and Alibaba Cloud accounts.
Scenario	Manage DMS users.
Procedure	In the top navigation bar of the DMS console, choose System Management > User and manage DMS users.
Limit	The operator must assume the administrator role.

13.4.10.3. Task management

Category	Description
Feature	This feature allows you to centrally manage tasks in DMS.
Scenario	Create tasks, stop running tasks, delete tasks, restart failed tasks, and view historical tasks.
Procedure	In the top navigation bar of the DMS console, choose System Management > Task . On the page that appears, manage tasks that are created by different tickets, for example, create or modify a task.
Limit	The operator must assume the administrator or DBA role.

13.4.10.4. Security rules

Category	Description
Features	This feature allows you to perform fine-grained instance-level access control on operations, development specifications, development processes, and approval processes.
Scenario	Create instance-level development specifications, development processes, and approval processes.
Procedure	In the top navigation bar of the DMS console, choose System Management > Security Rules and configure the required information. The information includes the approval process and related approvers.

Category	Description
Limit	• The operator must assume the administrator or DBA role. • The control mode of the related instance must be secure collaboration.

13.4.10.5. Approval processes

Category	Description
Feature	This feature allows you to specify approval processes, the approval nodes of each approval process, and the approvers of each approval node.
Scenario	Create a custom approval process.
Procedure	In the top navigation bar of the DMS console, choose System Management > Security > Approval Processes , create an approval process, and then specify the required information, such as approvers.
Limit	• The operator must assume the administrator or DBA role. • The control mode of the related instance must be security collaboration.

13.4.10.6. Operational logs

Category	Description
Feature	This feature allows you to view the logs that record the behaviors of all users.
Scenario	Audit the operations of a database, table, or operator in a specified period of time.
Procedure	In the top navigation bar of the DMS console, choose System Management > Operation audit and query the required information.
Limit	The operator must assume the administrator or DBA role.

13.4.10.7. IP whitelisting

Category	Description
Feature	This feature allows you to control access from source IP addresses to Data Management (DMS).
Scenario	Limit the range of IP addresses that can access DMS.
Procedure	In the top navigation bar of the DMS console, choose System Management > Security > Access IP Whitelists and specify whether to enable the feature. If this feature is enabled, you can specify CIDR blocks in the IP whitelist.

Category	Description
Limit	The operator must assume the administrator or DBA role.

13.4.10.8. Sensitive data management

Category	Description
Feature	This feature allows you to centrally manage fields that are labeled as sensitive or confidential. You can specify de-identification algorithms for these fields, for example, mask some specified positions, or replace some specified characters.
Scenario	Manage de-identification algorithms, for example, show only the last four digits of a mobile phone number or ID card.
Procedure	In the top navigation bar of the DMS console, choose System Management > Security > Sensitive Data and specify the required control method and threshold.
Limit	- The operator must assume the security administrator, database administrator (DBA), or administrator role. - The destination database must be a relational or analytic database. - An instance is in Secure Collaboration mode.

13.4.10.9. Row-level sensitive data management

Category	Description
Feature	This feature allows you to grant row-specific permissions to users. This applies to rows that are included in the same table.
Scenario	View user-specific data. For example, if you need to view the details of a chain enterprise, you can view the details of the region only for which you are responsible and cannot view the details of other regions.
Procedure	In the top navigation bar of the DMS console, choose System Management > Security > Sensitive Data . Then, click the Row Level Security tab on the left side of the page and set the required resources to be managed. These resources include the databases, tables, fields, and values of fields.
Limit	- The operator must assume the security administrator, database administrator (DBA), or administrator role. - The destination database must be a relational or analytic database. - An instance is in Secure Collaboration mode.

13.4.10.10. Configuration management

Category	Description
Feature	This feature allows you to manage global parameters that apply to Data Management. For example, you can specify a base threshold that includes the maximum number of rows that can be queried each day, or the maximum number of queries that can be issued each day.
Scenario	Create global control policies.
Procedure	In the top navigation bar of the DMS console, choose System Management > Configuration and specify the required control method and threshold.
Limit	The operator must assume the administrator role.

13.5. Terms

This topic describes the terms that are used in the Data Management (DMS) documentation.

System role

DMS provides the following roles: common users, security administrators, database administrators (DBAs), and administrators. The features that are supported by each role are different. For more information, see the Features of each role topic of *DMS User Guide*.

- Common users can perform multiple operations on databases, for example, query and change data, and view and change schemas. Common users can be engineers from different departments in an enterprise, for example, R&D, testing, product, operations, and data analysis.
- Security administrators can perform multiple operations, for example, classify security levels for data fields and audit user operations. Security administrators can be engineers from different departments in an enterprise, for example, audit and security.
- DBAs can perform multiple operations, for example, manage and maintain database instances, create development specifications and processes for databases, and execute tasks. DBAs can be engineers from different departments in an enterprise, for example, database processing and O&M.
- By default, an Apsara Stack tenant account assumes the administrator role. This role cannot be revoked. You can specify a RAM user or an account that is added to the current tenant as an administrator. No limit is set on the number of administrators.

 **Note** Administrators are specified at the admin stage of an approval process.

Resource role

Role	Description	Permission

Role	Description	Permission
Instance owner	- Each instance has only one owner. - For an ApsaraDB instance, the default instance owner is the Alibaba Cloud account that is used to create the ApsaraDB instance. For a database instance that is not an ApsaraDB instance, the default instance owner is the Alibaba Cloud account or RAM user that is used to add the database instance to DMS. - Administrators or the owner of an instance can transfer the ownership of the instance to another account.	- The owner of an instance can control access to the instance, for example, grant the required permissions to the instance or revoke these permissions. - The owner of an instance can query all the data in the databases of the instance, except for data in sensitive or confidential fields. The owner can also submit tickets to perform operations on data and schemas in the instance without the need to request the related permissions.
Database owner	- Each database has up to three owners. When the data dictionary of a database is synchronized for the first time, the DBA of the instance that owns the database becomes an owner of the database. - DBAs, administrators, or the owner of a database can perform multiple operations on other database owners, for example, add or remove a database owner, or transfer the ownership of the database to another account. - A DMS user can request to become a database owner by submitting a ticket.	- The owner of a database can control access to the database, for example, grant or revoke permissions on the database or the tables in the database. - The owner of a database can query all the data in the database, except for data in sensitive or confidential fields. The owner can also submit tickets to perform operations on data and schemas in the database without the need to request the related permissions. - Database owners are identified by the system and assigned to the owner nodes in approval processes.
Table owner	- Each table has up to three owners. By default, the owners of a table are the owners of the database to which the table belongs. - DBAs, DMS administrators, and the owner of a table can add or remove an owner of the table. These users can also transfer the ownership of the table from an existing owner to another user. - A DMS user can request to become a table owner by submitting a ticket.	The owner of a table can manage permissions on the table, for example, grant or revoke permissions on the table. The owner of a table can query all the data in the table, except for data in sensitive or confidential fields.

Role	Description	Permission
DBA	- Each instance has only one DBA. - DBAs and DMS administrators can manage instance DBAs.	- The DBA of an instance can view user permissions on the instance, and grant or revoke permissions on the databases and tables in the instance. - The DBA of an instance can query all the data in the databases of the instance. However, this excludes data in sensitive or confidential fields. The DBA can also submit tickets to perform operations on data and schemas in the instance without the need to first apply for permissions. - Instance DBAs are identified by the system and assigned to the DBA nodes in approval processes.

Control mode

DMS provides the following control modes: flexible management, stable change, and security collaboration. Control modes function at the instance level. In other words, you can specify the control mode of each instance.

Control mode	Description	Scenario	Logon method
Flexible management	This control mode allows you to manage the visualized data and schemas of multiple types of databases. It also provides a variety of data management solutions. This simplifies the use of databases and facilitates management.	- Database instances do not require strict control. - Database instances are used by a single user.	A database account and the related password.
Stable change	- This control mode provides multiple solutions to ensure database reliability. These solutions allow you to change data without the need to lock the related table or schema. - All features that are included in the flexible management control mode are available.	- Database instances require a high level of availability. This ensures that these database instances function as expected for an extended period of time. - Database instances are used by a small-sized group that includes multiple users.	A database account and the related password.

Control mode	Description	Scenario	Logon method
Security collaboration	- This control mode provides multiple solutions to ensure data security. These solutions include fine-grained access control at the database, table, or field level and sensitive data management. - This control mode allows you to produce enterprise-specific database DevOps solutions through custom design specifications and approval processes. - All features that are included in the flexible management and stable change control modes are available.	- Ensure the data security of database instances. - Implement strict access control over development or change workflows. - Manage compliance for enterprises.	Logon-free through authorization.

Routing algorithm

Definition: In some cases, you need to manage table shards or logical tables in the SQLConsole, change data, or export data. To perform these operations on logical tables and obtain more accurate results through the specified conditions, you can configure routing algorithms. This improves the efficiency of operations.

Note

- If you do not configure routing algorithms, operations on a logical table traverse each physical table that is related to the logical table. However, this significantly increases the time that is required for data processing.
- Routing algorithms consists of routing fields and algorithms.

Routing algorithms are used in the following scenarios:

- Query the data of table shards.
- Change the data of table shards.
- Export the data of table shards.

If you need to create or modify a routing algorithm, you can find the related logical table in the required database. Then, you can edit the routing algorithm.

Field-specific security levels

Definition: Field-specific security levels are applied based on the type of business data that is stored in databases. For example, some fields contain mobile phone numbers or ID card numbers. These numbers are considered to be as sensitive data. The values of these fields are not returned by common queries except for specific cases.

Field-specific security levels are used in the following scenarios:

- When you query data in the SQLConsole, the values of sensitive or confidential columns to which you

have no access are displayed as **. This avoids the exposure of sensitive data.

- If you need to access sensitive or confidential columns when you query, export, or change data, you must apply for the required permissions.
- If sensitive or confidential columns are accessed when you export or change data, you can configure different approval processes for DBAs and administrators.

Set the security level of a field based on the following rules:

Table schema: Modify the security level of a field when you view the related table schema.

- If you request an increase for the security level of a field, the new level takes effect immediately after you submit the request.
- If you request a decrease for the security level of a field, the new level will not take effect until the request has passed the approval process that is configured by a DBA or administrator.

Logical database

Definition:

- When your business reaches a specific scale, you need to balance workloads through database sharding and table sharding. This ensures smooth business expansion under high business pressure.
- If the number of databases is equal to or greater than one, take note of the following items:
 - If the number of databases is greater than one, you must set limits on the number of databases and suffix format of database names.
 - In most cases, the number of databases is an exponential power of 2.
 - A database name is suffixed by _<xxxx>. xxxx is a four-digit number. The number starts from 0000 and increments by 1 each time.
 - If the number of databases is 1, only a single database that has table shards is configured as a logical database.

Typical scenarios:

- A single database has table shards.
- Database shards have table shards.
- Database shards do not have table shards. It indicates that each database shard has only one table.

Usage:

- You can perform multiple operations on physical databases or logical databases. These operations include querying data in the SQLConsole, designing schemas, exporting data, and changing data.

If you select logical databases, you can perform operations on table shards or logical tables with ease. You can write SQL statements for a table shard or logical table in the same way you write SQL statements for a single table.

- To request the permissions on a data source, you can request only the permissions on the connected logical database. After the request succeeds, you can access all the physical databases that are related to the logical database.

Procedure:

On the Databases page, find the required physical database and choose **More** > Configure Logical Database. In the dialog box that appears, create a logical database through a data owner or DBA.

Logical table

Definition:

- When your business reaches a specific scale, you need to balance workloads through database sharding and table sharding. This ensures smooth business expansion under high business pressure.
- If the number of tables is greater than 1, workloads are evenly distributed across logical databases.
 - In most cases, the number of tables is an exponential power of 2.
 - A table name is suffixed by _<xxxx>. xxxx is a four-digit number. The number starts from 0000 and increments by 1 each time.
 - The number of tables must be evenly divided by the number of databases. For example, if the number of tables is 1,024 and the number of databases is 32, each database obtains 32 tables. If the number of tables is 1,024 and the number of databases is 33, these tables cannot be evenly divided. Therefore, no logical table can be created.

Typical scenarios:

- A single database has table shards.
- A single physical database has table shards.
- Database shards have table shards. This is the most common case.
 - Physical tables are evenly divided by the specified number of physical databases. The number increments by 1 for each new physical table that is added to a physical database.
 - The names of physical tables that are distributed to each database are the same. For example, each database has 12 tables that are numbered from 01 to 12.
- Database shards do not have table shards. It indicates that each database shard has only one table.

 **Note** Common case: Each database shard has a physical table whose name is the same as the name of other physical tables.

Usage:

- If you manage local tables in the preceding scenarios, you must perform operations on logical tables in a logical database.
- Request the permissions for a data source.

After you request the permissions on a logical table and the request succeeds, you can access all the physical tables that are related to the logical table.

- Query data in the SQLConsole.
- Export data.
- Change data.
- You can perform the preceding operations on a specified logical database. Then, you can use a logical table in the same way you use a single table.

Procedure:

- Initialize

After you configure a logical database, the system creates a logical table based on the schema of a physical table in the local database. You cannot modify the preceding settings.

- Extraction rules
 - A physical table can be specified in the settings of only one logical table.

- All physical tables that correspond to the same logical table must have the same schemas (including the field names and field types). Otherwise, the physical tables cannot be aggregated. This rule allows the system to send you alert notifications when data inconsistency occurs.

- Create

If the existing logical table is not created by DMS, you can perform the following steps to create a logical table: On the Databases page, find the required logical database and choose More > Extract Logical Tables.

FAQ:

- Q: If a logical table exists in a logical database, I do not need to create a physical table. However, the logical table is excluded from the logical table list. In this case, what can I do?
- A: You can perform the following steps to create a logical table: Search for the required logical database and choose Actions > More > Extract Logical Tables.

Metadata

Destination:

Metadata includes the predefined data of a database. Collected metadata includes the following details:

- The database name, database character set,
- table names, size of each table, number of rows that are included in each table, character set of each table, fields of each table, indexes of each table, description of each table
- type of each field, precision of each field, and description of each field.

 **Note** Note: All metadata is collected from the internal data of databases, for example, information_schema. Therefore, some information may be inaccurate, for example, the table size and the number of rows in a table. In this case, the information can be used for reference only.

Scenarios:

View and export schemas.

- After you enter the keyword of a database name in the top navigation bar, you can use one of the following methods to go to the database search result page:
 - Press Enter. Note: Do not move the pointer over a record in the auto-completion list.
 - Click Search in the auto-completion list.
 - Click the Magnifier icon next to the search box.
- After you enter the keyword of a database name in the top navigation bar, you can use one of the following methods to go to the database table list page:
 - Click the row of the required database in the auto-completion list.
 - Move the pointer over the row of the required database in the auto-completion list and press Enter.
- On the database search result page, use one of the following methods to go to the database table list page:
 - Click Details in the row of the required database.
 - Choose More > View Table Details in the row of the required database.

- Double-click the row of the required database.

Use one of the preceding methods to go to the **database search result page**. Find the required database and choose More > Export Table Creation Statements > (CREATE TABLE statements) or Export Schema (in Excel format).

 **Note** The data of all tables in the database is exported

Use one of the preceding methods to go to the **database table list page**.

- In the upper-right corner of the tab, click Export. From the menu that appears, select All Table Creation Statements in Database (CREATE TABLE statements) or All Schemas in Database (in Excel format).

 **Note** The data of all tables in the database is exported.

- In the row of the specified table, choose More > Export Table Creation Statements (CREATE TABLE statements) or Export Schema (in Excel format). Only the data of a single physical table is exported. If you only need to view the schema of a table without the need to export the table schema, click the row of the required table. The fields, field attributes, indexes, index attributes of the table are displayed.

Synchronization mechanism:

When you register an instance in the DMS console, the system collects the data dictionary of the instance in full data mode. However, if a DDL operation is performed outside the system, the system cannot identify this operation in real time. In this case, you need to synchronize metadata to the system once.

- Method 1: If you are a common user, click Sync Metadata in the upper-right corner of the database table list page.
 - If the database is physical, only the data of the current database is synchronized.
 - If the database is logical, the data of all physical databases that are related to the logical database is synchronized in batches.
- Method 2: If you are a database administrator or administrator, choose System Management > Instance, find the required instance, and then click Sync Now.
- Method 3: If you are an administrator, choose System Management > Configuration and find the required configuration. Then, create a **scheduled task** to synchronize the metadata of all instances on a daily basis.

 **Note** Note: For a new database, you can use only Method 2 to synchronize metadata. If you want to synchronize metadata after table data in an existing database is changed, you can use Method 1 or Method 2.

Data owner

Definition: an operator that is responsible for the data of a database or table. The operator controls access to the database or table from other users.

Usage:

- A data owner can be specified as an approval node in the approval process of each functional

module on the System Management > Security Rules page.

- A data owner can grant the permissions of a database or table to a user or revoke these permissions.

Procedure:

- Passive: The data owner of a database or table can be specified by a DBA or the original data owner of the database or table.
- Active: The operator who creates a database or table can request to become the data owner of the database or table.

 **Note** An operator can request to become the data owner of a database or table. After the request has passed the approval process that is specified by the related instance, the data owner is specified.

Security rule

This section describes the following three core terms: security rule, approval process, and approval node.

- Approval node
 - The system provides the following default dynamic nodes. You cannot delete or edit these nodes.
 - Admin: the administrators of the system. If the system has multiple administrators, only one is required to participate in an approval process.
 - DBA: the database administrator (DBA) of an instance. The administrator or DBA who adds the instance is assigned to this node by default. You can also assign another DBA in the system to this node.
 - Owner: the data owner of a database. The data owner of a database or table can be configured by the DBA when the DBA registers an instance and collects the data dictionary. In addition, the original data owner of a database or table can configure another user as the new data owner. A user can also request to become the data owner of a database or table.
 - The system supports the following custom approval nodes. You can delete or edit these nodes as required.
 - You can create custom approval nodes and the related approvers based on the default system nodes. The following list includes commonly used nodes:
 - Test engineer
 - Data security engineer
 - R&D TL
 - Architect
 - Note: You can specify a person that is specific to a business unit as an approval node, for example, a test engineer or R&D TL.
- Approval process

- The system provides the following default dynamic approval processes. You cannot delete or edit these approval processes.
 - Admin: A ticket can be approved only by an administrator.
 - DBA: A ticket can be approved only by a DBA.
 - Owner: A ticket can be approved only by a data owner.
 - Owner > DBA: A ticket is approved by a data owner and DBA in sequence.
 - Owner > DBA > Admin: A ticket is approved by a data owner, DBA, and DBA in sequence.
- The system supports the following custom approval processes. You can delete or edit these approval processes as required.
 - You can combine dynamic nodes of the system with custom approval nodes to create a custom approval process. The following list includes commonly used custom approval processes:
 - Owner > Data security engineer: A ticket is approved by a data owner and data security engineer in sequence.
 - Owner > Test engineer: A ticket is approved by the data owner and test engineer in sequence.
 - Note: You can use a combination of various approval nodes to form a custom approval process based on your business requirements. This custom approval process is specific to your business line.
- Security rule
 - By default, the system provides security rules that are based on the following three levels: high, medium, and low. For more information about how to apply a rule to a module, see the related documentation. This section describes the features of each module. You can edit the default security rules. However, you cannot delete these security rules.
 - SQLConsole:
 - Specifies whether you can run data manipulation language (DML) statements. You can set a threshold for the number of affected rows. If the threshold is exceeded, DML statements cannot be run.)
 - Specifies whether you can run data definition language (DDL) statements. You can set a threshold for the size of a table. If the threshold is exceeded, DDL statements cannot be run.
 - Specifies whether you can perform high-risk DDL operations, such as deleting tables and deleting fields.
 - Specifies whether you can run other SQL statements.

- Data change

- Specifies whether you can run DMS statements. You can set a threshold for the number of affected rows. If the threshold is exceeded, DML statements cannot be run.

 **Note** If yes, you can specify whether an approval process is required and what approval process is required.

- Specifies whether you can run DDL statements. You can set a threshold for the size of a table. If the threshold is exceeded, DDL statements cannot be run.

 **Note** If yes, you can specify whether an approval process is required and what approval process is required.

- Specifies whether you can perform high-risk DDL operations, such as deleting tables and deleting fields.

 **Note** If yes, you can specify whether an approval process is required and what approval process is required.

- Specifies whether you can run other SQL statements.

 **Note** If yes, you can specify whether an approval process is required and what approval process is required.

- Data export

Specifies whether an approval is required to export data.

- If yes, you can set a threshold for the number of rows that can be exported each time.

 **Note** Based on different thresholds, you can specify whether an approval process is required and what approval process is required.

- When the rows to be exported include sensitive data, you can create a custom approval process and set the approval process as the most rigorous process. In this case, you do not need to be concerned about how many rows are exported.

- Permission application
 - Permissions on databases or tables

Specifies whether an approval process is required and what approval process is required.
 - Permissions on databases or tables
 - Permissions on sensitive columns
 - Permissions on confidential columns
- Data owner

Specifies whether an approval process is required and what approval process is required.

 - Approval process when a data owner already exists
 - Approval process when no data owner exists
- Sensitive levels. After you increase a security level, the new security level immediately takes effect.

Specifies whether an approval process is required and what approval process is required.

 - Decrease a security level from confidential to sensitive.
 - Decrease a security level from confidential to inner.
 - Decrease a security level from sensitive to inner.
- Custom security rules. You can delete or edit these security rules.

You can use a combination of multiple approval processes to configure security rules that meet the requirements of a business line. This allows you to implement flexible management and perform an operation audit on your business.

 **Note** Each instance conforms to one security rule. You can implement strict control on a test environment. You can also implement loose control on a production environment in a backend system. This allows you to implement flexible on-demand management.

Task orchestration

- A task is a node that can complete a functional operation. DMS supports the following task types: single-instance SQL, cross-database SQL, and data synchronization.
- A task flow is a task group that consists of multiple task nodes. You can manage task flows based on a directed acyclic graph (DAG).
- The following roles are supported.
 - Owner: Only an owner of a task flow has the permissions to edit the task flow and the related task settings. The owner can also perform a dry run on tasks and receive task alert notifications.
 - Stakeholder: A stakeholder of a task flow has the permissions to view the task flow and the related task settings. The stakeholder can perform a dry run on tasks. However, the stakeholder does not have the permissions to edit the task flow and task settings.
 - Administrator and DBA: In addition to the permissions of a stakeholder, an administrator or DBA has the permissions to edit and reassign the owner of a task flow.
- Execution permission: specifies whether a task flow can be run based on the related database and table permissions that are granted to the owner of the task flow. An operator that assumes one of the following roles can run a task flow: owner, stakeholder, administrator, and DBA.

 **Note** If the owner does not have permissions on a database or table, the task flow still fails even if an operator that runs the task flow has access to the related database or table.

- **Running instance:** A running instance is generated each time a task flow is run. You can view or manage instances on the Instance page.
- **Running time:** indicates the actual time when a task is run. The time is displayed in the UTC+8 format.
- **Business time:** By default, the business time is one day before the running time.
- **Custom variable:** To run SQL tasks at a different time to obtain different results, you can create custom variables. DMS allows you to specify custom variables for SQL tasks. A custom variable indicates an offset based on a business time.

DBLink

The cross-database query feature provides database links. A database link can direct you to an arbitrary database instance. A database link is also used as the alias of a database instance. Each database link corresponds to a database instance. If you are using MySQL, a database link includes the IP address and port of a MySQL database.

14. Server Load Balancer (SLB)

14.1. What is SLB?

This topic provides an overview of Server Load Balancer (SLB). SLB distributes inbound network traffic across multiple Elastic Compute Service (ECS) instances that function as backend servers based on forwarding rules. You can use SLB to improve the responsiveness and availability of your applications.

Overview

After you attach ECS instances that are deployed in the same region to an SLB instance, SLB uses virtual IP addresses (VIPs) to virtualize these ECS instances into backend servers in a high-performance server pool that ensures high availability. Client requests are distributed to the ECS instances based on forwarding rules.

SLB checks the health status of the ECS instances and automatically removes unhealthy ones from the server pool to eliminate single points of failure (SPOFs). This enhances the resilience of your applications.

Components

SLB consists of three components:

- SLB instances

An SLB instance is a running SLB service entity that receives traffic and distributes traffic to backend servers. To get started with SLB, you must create an SLB instance, configure at least one listener for the SLB instance, and attach at least two ECS instances to the SLB instance.

- Listeners

A listener checks client requests and forwards them to backend servers. It also performs health checks on backend servers.

- Backend servers

ECS instances are used as backend servers to receive distributed requests. You can add ECS instances one by one to the server pool, or use vServer groups or primary/secondary server groups to add and manage multiple ECS instances at a time.

Benefits

- High availability

SLB is designed with full redundancy that prevents SPOFs and supports zone-disaster recovery.

SLB can be scaled based on application loads and can provide continuous services during traffic fluctuations.

- High scalability

You can increase or decrease the number of backend servers to adjust the load balancing capability of your applications.

- High cost efficiency

SLB can save 60% of load balancing costs compared with traditional hardware solutions.

- Outstanding security

You can integrate SLB with Apsara Stack Security to defend your applications against DDoS attacks of up to 5 Gbit/s.

- High concurrency

An SLB cluster supports hundreds of millions of concurrent connections. A single SLB instance supports tens of millions of concurrent connections.

14.2. High availability

This topic describes the high availability of Server Load Balancer (SLB). SLB provides a high-availability architecture based on system design and product configurations. To meet your business requirements, you can use SLB together with services such as Apsara Stack DNS to implement geo-disaster recovery.

High availability of SLB

SLB instances are deployed in clusters to synchronize sessions and protect backend servers from single points of failure (SPOFs). This improves redundancy and ensures service stability. Layer 4 SLB uses the Linux Virtual Server (LVS) and Keepalived software to balance loads, whereas Layer 7 SLB uses Tengine. Tengine, a web server project launched by Taobao, is based on NGINX and adds advanced features that are designed for high-traffic websites.

Requests from the Internet are forwarded to LVS clusters based on equal-cost multi-path (ECMP) routing. In an LVS cluster, each LVS machine uses multicast packets to synchronize sessions with other LVS machines. This way, sessions are synchronized among all machines in the LVS cluster. LVS clusters also perform health checks on Tengine clusters. To ensure the availability of Layer 7 SLB, unhealthy devices are removed from Tengine clusters.

Best practices:

You can use session synchronization to prevent persistent connections from being affected by server failures within a cluster. However, for short-lived connections, server failures in the cluster may affect user access. This also occurs when a connection does not trigger session synchronization because the three-way handshake is not complete. To prevent session interruptions due to server failures within the cluster, you can add a retry mechanism to the service logic. This reduces the impact of server failures on user access.

High-availability solution with one SLB instance

Apsara Stack allows you to deploy SLB instances across multiple zones in different regions. You can deploy an SLB instance in primary/secondary zone mode. This ensures the high availability of the SLB instance. If the primary zone fails or becomes unavailable, a failover is triggered to redirect requests to servers in the secondary zone in about 30 seconds. After the primary zone recovers, traffic is automatically switched back to servers in the primary zone.

 **Note** Zone-disaster recovery is implemented between the primary and secondary zones. SLB implements failovers only when the entire SLB cluster within the primary zone is unavailable due to factors such as power outage and optical cable failures. A failover is not triggered when a single instance in the primary zone fails.

Best practices:

1. We recommend that you create SLB instances in regions that support primary/secondary zone deployment for zone-disaster recovery.
2. You can determine the primary and secondary zones for an SLB instance based on the distribution

of Elastic Compute Service (ECS) instances. Select the zone where most ECS instances are deployed as the primary zone to minimize access latency.

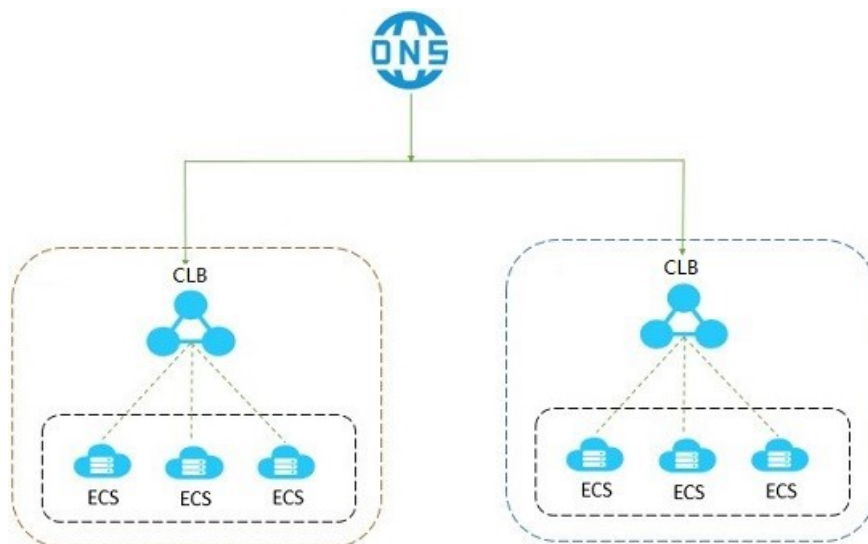
However, we recommend that you do not deploy all ECS instances in the primary zone. You must deploy a small number of ECS instances in the secondary zone. This way, requests can be redirected to backend servers in the secondary zone when the primary zone becomes unavailable.

High-availability solution with multiple SLB instances

If you require extremely high availability, the high-availability solution with one SLB instance may be insufficient for your needs. If an SLB instance becomes unavailable due to network attacks or invalid configurations, failovers between the primary and secondary zones are not triggered. To avoid such issues, you can create multiple SLB instances. Then, you can use Apsara Stack DNS to schedule requests, or use a global load balancing solution to achieve cross-region backup and disaster recovery.

Best practices:

You can deploy SLB instances and ECS instances in multiple zones within a region or across multiple regions, and schedule requests by using Apsara Stack DNS.



High-availability solution with backend ECS instances

SLB performs health checks to verify the availability of backend ECS instances. The health check feature improves the availability of frontend services by minimizing the impacts of ECS instance exceptions on the services.

After you enable the health check feature, when an unhealthy ECS instance is detected, SLB distributes new requests to other healthy ECS instances. After the ECS instance recovers and is confirmed healthy, SLB automatically sends requests to the ECS instance. For more information about the health check feature, see *Health check overview* in *User Guide*.

Best practices:

To perform health checks, make sure that the health check feature is enabled and properly configured. For more information, see *Configure health checks* in *User Guide*.

14.3. Architecture

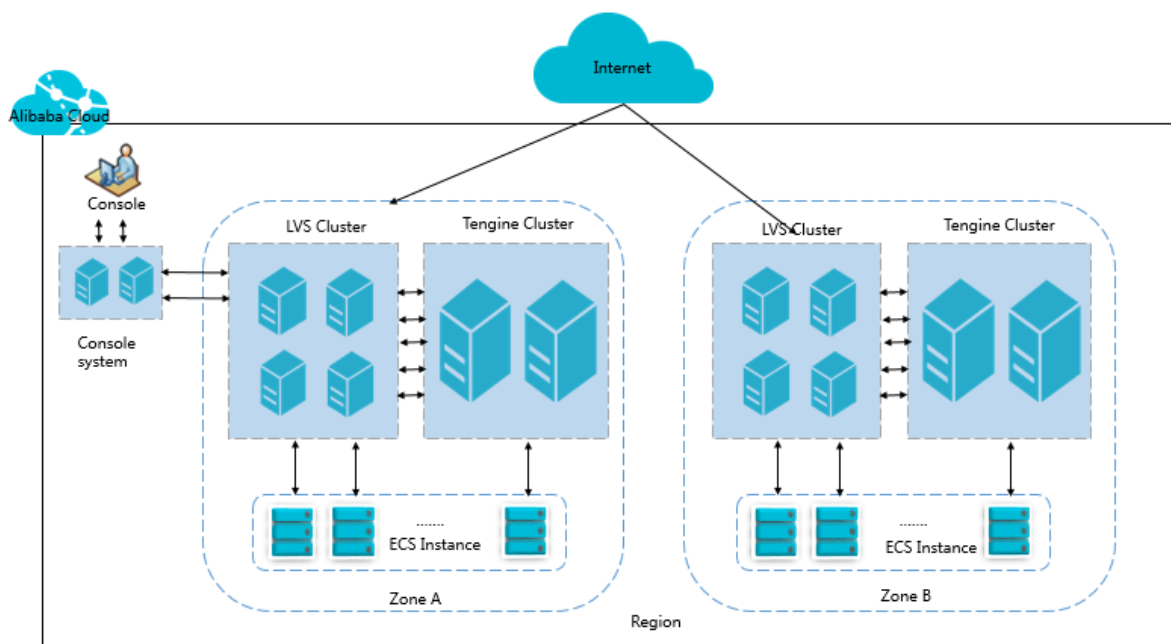
This topic describes the architecture of Server Load Balancer (SLB). SLB instances are deployed in clusters to synchronize sessions and protect backend servers from single points of failure (SPOFs). This improves redundancy and ensures service stability. SLB supports load balancing at Layer 4 for TCP and UDP traffic, and at Layer 7 for HTTP and HTTPS traffic.

SLB uses SLB clusters to forward client requests to backend servers and receives responses from backend servers over the internal network.

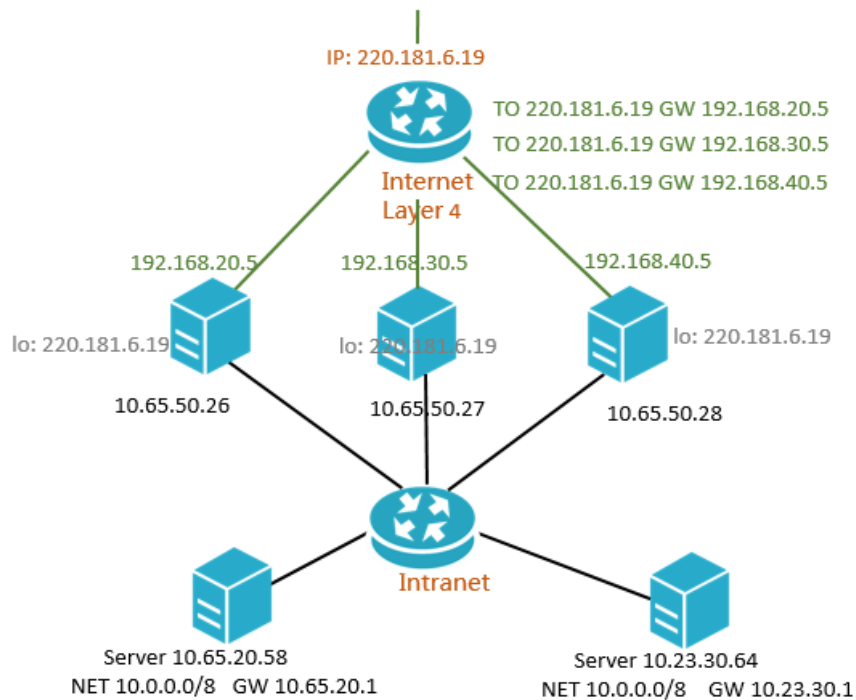
SLB design

SLB provides load balancing services at Layer 4 and Layer 7.

- Layer 4 SLB uses the open source Linux Virtual Server (LVS) and Keepalived software to balance loads, and adapts the software to meet cloud computing requirements.
- Layer 7 SLB uses Tengine to balance loads. Tengine, a web server project launched by Taobao, is based on NGINX and adds various advanced features that are designed for high-traffic websites.

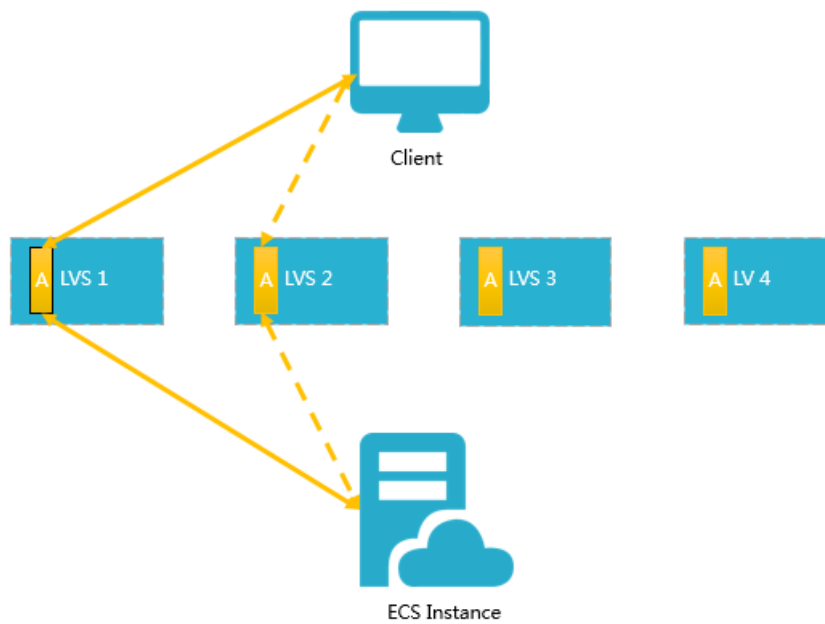


In each region, Layer 4 SLB runs in an LVS cluster that consists of multiple LVS machines, as shown in the following figure. This cluster deployment mode enhances the availability, stability, and scalability of load balancing services in abnormal cases.



In an LVS cluster, each LVS machine uses multicast packets to synchronize sessions with other LVS machines. This way, sessions are synchronized among all the machines in the LVS cluster. For example, after the client sends three packets to the server, Session A established on LVS1 is synchronized with other LVS machines, as shown in the following figure. Solid lines indicate the active connections. Dashed lines indicate that requests are sent to other normally working machines such as LVS2 if LVS1 fails or is being maintained. This way, you can perform hot upgrades, machine failure maintenance, and cluster maintenance without affecting the services of your applications.

Note If a connection is not established because the three-way handshake is not complete, or if a connection has been established but session synchronization is not triggered during a hot upgrade, your services may be interrupted. In this case, you must reinitiate a connection request from the client.



14.4. Features

This topic describes the features of Apsara Stack Server Load Balancer (SLB). SLB provides load balancing at Layer 4 and Layer 7. It supports features such as health checks, session persistence, and domain name-based forwarding rules to ensure the high availability of backend services.

In the following table, Y indicates that the feature is supported, and N indicates that the feature is not supported.

Feature	Layer 4 SLB	Layer 7 SLB
Scheduling algorithms SLB supports the round-robin (RR), weighted round-robin (WRR), and consistent hashing scheduling algorithms.	Y	Y Note Layer 7 SLB does not support the consistent hashing scheduling algorithm.
Health checks SLB checks the health status of backend servers. When an unhealthy backend server is detected, SLB stops distributing inbound traffic to the backend server. Network traffic is distributed to other backend servers that work as expected.	Y	Y

Feature	Layer 4 SLB	Layer 7 SLB
Session persistence SLB supports session persistence. After session persistence is enabled, SLB can distribute requests from a client during a session to the same backend server.	Y	Y
Access control SLB uses whitelists to control access to your applications.	Y	Y
High availability SLB distributes inbound traffic to backend servers that are deployed in different zones. In most regions, Apsara Stack allows you to deploy SLB instances in primary/secondary zone mode across multiple zones. If the primary zone fails, a failover is triggered to redirect requests to servers in the secondary zone.	Y	Y
Security You can integrate SLB with Apsara Stack Security to defend your applications against DDoS attacks of up to 5 Gbit/s.	Y	Y
Network types Apsara Stack provides Internet-facing and internal-facing SLB instances. To process network traffic within a virtual private cloud (VPC), you can create an internal-facing SLB instance. To process network traffic from the Internet, you can create an Internet-facing SLB instance.	Y	Y
IPv6 Internet-facing SLB instances can forward requests from IPv6 clients.	Y	Y
Certificate management SLB manages certificates for HTTPS in a centralized way. You do not need to upload certificates to backend servers. Requests are decrypted on SLB instances before the requests are sent to backend servers. This reduces the CPU utilization on backend servers.	N	Y
WebSocket Secure and WebSocket WebSocket is an HTML5 protocol that provides full-duplex communication channels between clients and servers. You can use WebSocket to save server resources and bandwidth, and enable real-time communication.	N	Y

Feature	Layer 4 SLB	Layer 7 SLB
HTTP/2 HTTP/2 is the second major version of the HTTP protocol and is backward compatible with HTTP/1.x. In addition, HTTP/2 improves performance by optimizing the flow of content.	N	Y

14.5. Scenarios

Server Load Balancer (SLB) can be used to improve the availability and reliability of applications with high access traffic.

Balance the loads of your applications

You can configure listening rules to distribute heavy traffic among Elastic Compute Service (ECS) instances that are attached as backend servers to SLB instances. You can also use the session persistence feature to forward all requests from the same client to the same backend ECS instance to enhance access efficiency.

Scale your applications

You can add or remove backend ECS instances to scale the service capability of your applications based on your business requirements. SLB is applicable to both web servers and application servers.

Eliminate single points of failure

You can attach multiple ECS instances to an SLB instance. When ECS instances malfunction, SLB automatically isolates these ECS instances and distributes inbound requests to other healthy ECS instances. This ensures that your applications continue to run as expected.

Implement zone-disaster recovery (multi-zone disaster recovery)

To provide more stable and reliable load balancing services, Apsara Stack allows you to deploy SLB instances across multiple zones in different regions for disaster recovery. You can deploy an SLB instance in primary/secondary zone mode. If the primary zone fails or becomes unavailable, a failover is triggered to redirect requests to servers in the secondary zone in about 30 seconds. After the primary zone recovers, traffic is automatically switched back to servers in the primary zone. We recommend that you plan the deployment of backend servers based on your business requirements. We recommend that you add at least one ECS instance in each zone to achieve the highest load balancing efficiency.

ECS instances in different zones can be attached to an SLB instance, as shown in the following figure. In most cases, the SLB instance distributes inbound traffic to ECS instances in the primary zone (Zone A) and those in the secondary zone (Zone B). If Zone A fails, the SLB instance distributes inbound traffic only to ECS instances in Zone B. This deployment mode avoids service interruptions due to zone-level failures and reduces latency.

For example, you deploy all ECS instances in the primary zone (Zone A), and no ECS instances are deployed in the secondary zone (Zone B), as shown in the following figure. If the primary zone fails, your services are interrupted because no ECS instances are available in the secondary zone. This deployment mode achieves low latency but mitigates high availability.

Cross-region disaster recovery

You can deploy SLB instances in different regions and attach ECS instances of different zones within the same region to an SLB instance. You can use Apsara Stack DNS to resolve domain names to the endpoints of SLB instances in different regions for global load balancing purposes. When a region becomes unavailable, you can temporarily stop DNS resolution in the region without affecting user access.

14.6. Limits

This topic describes the quotas of Server Load Balancer (SLB) resources.

Item	Default limit
Limits on SLB instances	
Maximum number of listeners that can be configured for an SLB instance	500
Limits on certificates	
Maximum number of server certificates that can be uploaded in a region	1,000
Maximum number of client certificate authority (CA) certificates that can be uploaded in a region	1,000

14.7. Terms

This topic introduces the terms used in Server Load Balancer (SLB).

Term	Description
SLB	A network load balancing service provided by Apsara Stack. SLB distributes traffic across Elastic Compute Service (ECS) instances. SLB provides load balancing at Layer 4 and Layer 7.
SLB instance	A running SLB service entity. To get started with SLB, you must create an SLB instance.
endpoint	An IP address assigned to an SLB instance. The IP address can be either public or private based on the type of the SLB instance. You can resolve a domain name to a public IP address of an SLB instance to provide external services.
listener	A listener that defines how to forward inbound requests to backend servers. At least one listener must be configured for each SLB instance.
backend server	An ECS instance that receives client requests distributed by an SLB instance.
default server group	A group of ECS instances that process distributed requests. If a listener is not configured with a vServer group or primary/secondary server group, the listener forwards traffic to the backend servers in the default server group.

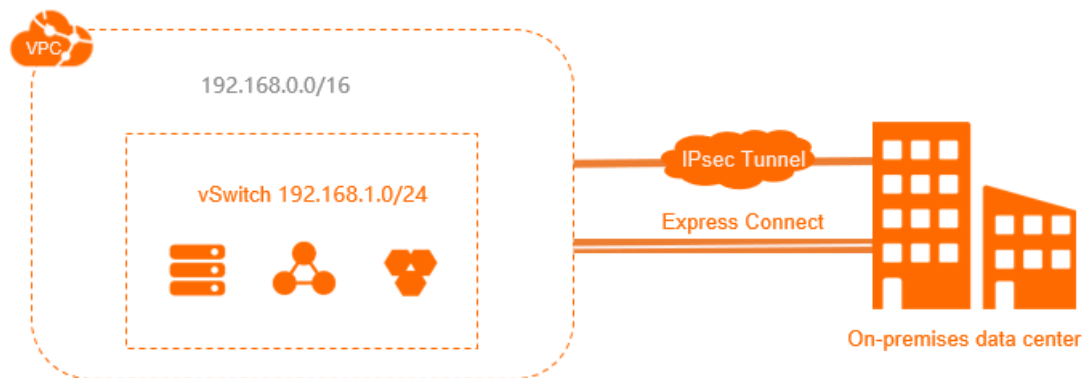
Term	Description
vServer group	A group of ECS instances that process distributed requests. You can create multiple vServer groups for different listeners of an SLB instance. This way, the listeners can forward traffic to the backend servers in different vServer groups.
primary/secondary server group	A server group that contains two ECS instances, which function as the primary server and the secondary server. If the primary server is detected unhealthy, new requests are automatically distributed to the secondary server.

15.Virtual Private Cloud (VPC)

15.1. What is a VPC?

A virtual private cloud (VPC) is a private network dedicated for your use. You have full control over your VPC. For example, you can specify the CIDR block and configure route tables and gateways. In a VPC, you can deploy Apsara Stack resources, such as Elastic Compute Service (ECS) instances, ApsaraDB RDS instances, and Server Load Balancer (SLB) instances.

Furthermore, you can connect your VPC to other VPCs or on-premises networks to create a custom network environment. This way, you can migrate applications to the cloud and extend data centers.



Components

Each Virtual Private Cloud consists of at least one private network segment, one router, and at least one switch.

- Private CIDR blocks

When you create a VPC and a vSwitch, you must specify the private IP address range for the VPC in CIDR notation.

You can use the standard private CIDR blocks listed in the following table and their subsets as CIDR blocks for your VPCs. For more information, see [网络规划](#). For more information, see The network planning section in *User Guide*.

CIDR blocks	Number of available private IP addresses (system reserved ones excluded)
192.168.0.0/16	65,532
172.16.0.0/12	1,048,572
10.0.0.0/8	16,777,212

- VRouter

A VRouter is a hub that connects all VSwitches in a VPC and serves as a gateway between the VPC and other networks. After a VPC is created, the system creates a VRouter for the VPC. Each vRouter is associated with a route table.

For more information, see [Overview](#).

For more information, see the Route table overview topic in *User Guide*.

- vSwitches

A vSwitch is a basic network component that connects different cloud resources in a VPC. After you create a VPC, you can create a vSwitch to divide your VPC into multiple subnets. vSwitches deployed in a VPC can communicate with each other over the private network. You can deploy your applications in vSwitches that belong to different zones to improve service availability.

For more information, see [Switch](#).

For more information, see the Create a vSwitch topic in *User Guide*.

15.2. Benefits

This topic describes the benefits of using VPCs.

High security

Each VPC has a unique tunnel ID, and each tunnel ID corresponds to a virtual network. Different VPCs are isolated by tunnel IDs:

- Similar to traditional networks, VPCs can also be divided into subnets. ECS instances in the same subnet use the same vSwitch to communicate with each other, while ECS instances in different subnets use VRouter to communicate with each other.
- VPCs are completely isolated from each other and can only be interconnected by mapping an EIP or a NAT IP address.
- ECS IP packets are encapsulated by using the tunneling technique. Therefore, information about the data link layer (layer-2 MAC address) of ECS does not go to the physical network. As a result, the layer-2 network between different ECS instances or between different VPCs is isolated.
- ECS instances in a VPC use security groups as firewalls to control traffic going to and from ECS instances. This is layer-3 isolation.

High flexibility

You can use security groups or whitelists to flexibly control traffic going to and from the cloud resources in a VPC.

Ease of use

You can quickly create and manage VPCs in the VPC console. After a VPC is created, the system automatically creates a VRouter and a route table for the VPC.

High scalability

You can create multiple subnets in a VPC to deploy different services. Additionally, you can connect a VPC to other VPCs or on-premises data centers to expand your network.

15.3. Architecture

Virtual Private Cloud (VPC) provides an isolated virtual network. This allows you to manage cloud resources in a secure environment based on tunneling technology. Each VPC is identified by a unique tunnel ID.

Background

The continuous development of cloud computing technologies leads to more demanding requirements for virtual networks, such as scalability, security, reliability, privacy, and robust connectivity performance. This scenario has hastened the birth of various virtualization network technologies.

Earlier solutions combined virtual and physical networks to generate a flat network architecture, such as large-scale Layer-2 networks. As the scale of virtual networks grew, earlier solutions faced more serious problems. A few notable problems include ARP spoofing, broadcast storms, and host scanning. Various network isolation technologies emerged to resolve these problems by completely isolating the physical networks from the virtual networks. One of the technologies utilizes VLAN to isolate users, but VLAN can support only up to 4096 users. It cannot support the large number of users in the cloud computing scenario.

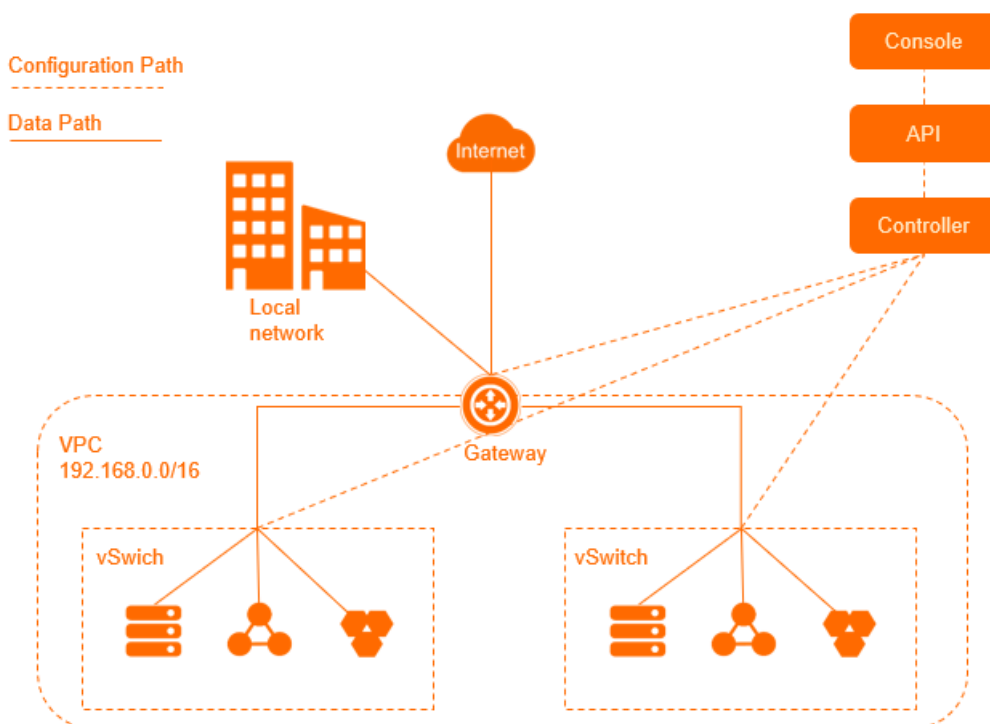
Principle

VPC provides an isolated virtual network to manage cloud resources in a secure environment based on tunneling technology. Each VPC is identified by a unique tunnel ID. Data packets are encapsulated with a unique tunnel ID and transmitted over the physical network between Elastic Compute Service (ECS) instances in a VPC. In different VPCs, ECS instances with different tunnel IDs are located on two different routing planes. Therefore, these ECS instances cannot communicate with each other.

Based on the tunneling and Software Defined Network (SDN) technologies, Alibaba Cloud has developed VPCs that are integrated with gateways and VSwitches.

Logic architecture

The following figure shows that a VPC consists of a gateway, a controller, and one or more VSwitches. The VSwitches and gateways form a key data path. By using a protocol developed by Alibaba Cloud, the controller distributes the forwarding table to the gateway and VSwitches to provide a key configuration path. In the overall architecture, the configuration path and data path are separated from each other. VSwitches are distributed nodes. The gateway and controller are deployed in clusters. Multiple data centers are built for backup and disaster recovery. Redundant links are provided for disaster recovery. This deployment mode improves the overall availability of the VPC.



15.4. Features

A Virtual Private Cloud (VPC) is a private network logically isolated from other virtual networks. This topic describes the features of VPCs.

Custom private networks

You can customize VPCs. When you create VPCs or VSwitches, you can specify private CIDR blocks for them. Furthermore, you can divide a VPC into multiple subnets and deploy services in different subnets to improve service availability.

Custom routes

You can add custom routes to the route table of a VPC to forward traffic to the specified next hops. The route table uses the longest prefix match algorithm for traffic routing. If one destination address may match more than one entry in the route table, the algorithm is used to select the entry with the longest subnet mask because it is the most specific route.

Varied connection methods

A VPC provides you with varied connection methods. You can connect a VPC to the public network, an on-premises data center, or another VPC.

- Connect a VPC to the public network

You can connect a VPC to the public network by associating an Elastic IP address with the VPC or configuring NAT Gateway, so that cloud services in the VPC can communicate with the public network.

- Connect a VPC to another VPC

You can connect a VPC to another VPC by creating a pair of router interfaces to enable high speed and secure communication between the VPCs over the internal network.

- Connect a VPC to an on-premises data center

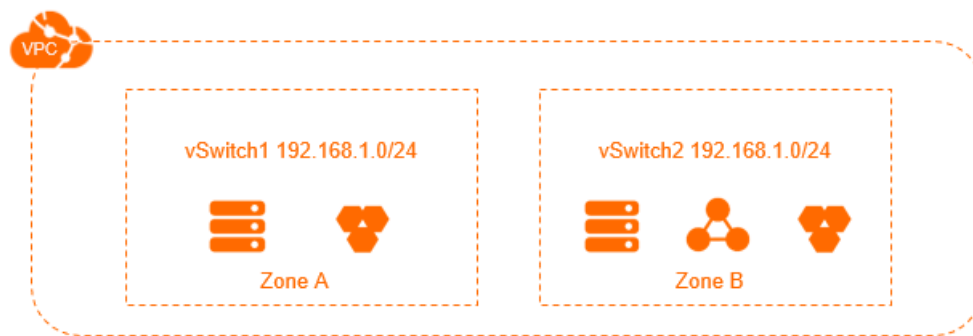
You can connect a VPC to an on-premises data center by using a leased line to smoothly migrate local applications to the cloud.

15.5. Scenarios

A virtual private cloud (VPC) is an isolated network environment that you can configure to meet your business requirements in different scenarios.

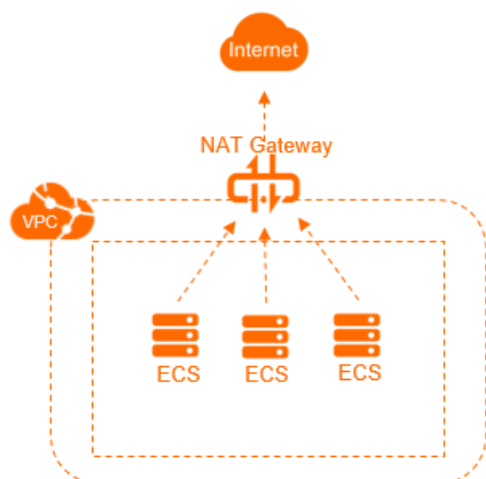
Host applications

Applications that are hosted in a VPC can provide external services. To control access to the applications over the Internet, you can create security group rules and whitelists. You can also isolate application servers from databases to implement access control. For example, you can deploy web servers in a subnet that allows access to the Internet, and deploy the databases in another subnet that denies access to the Internet.



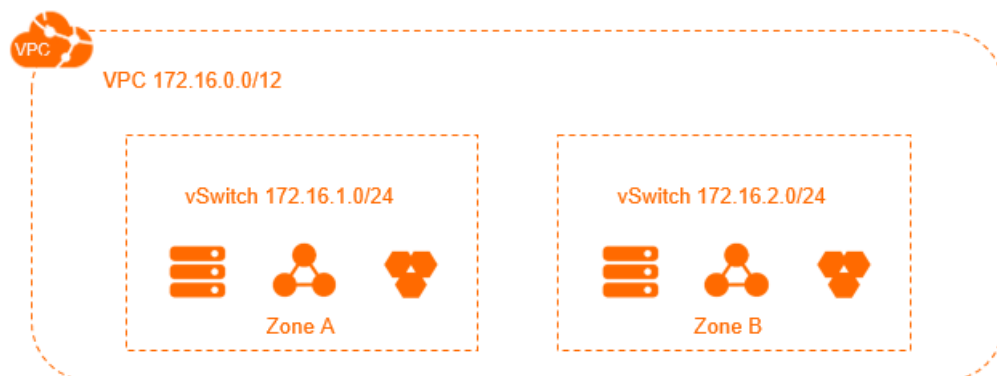
Host applications that require access to the Internet

You can host applications that require access to the Internet in a subnet of a VPC and use a NAT gateway to route network traffic. You can configure SNAT rules to allow instances in the subnet to access the Internet without the need to expose the private IP addresses. In addition, you can change the public IP address that is used to access the Internet at any time to prevent attacks from the Internet.



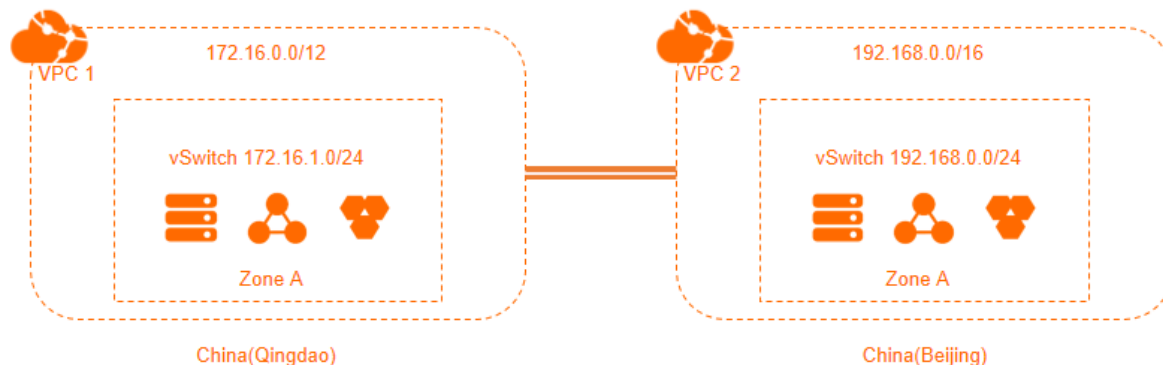
Implement cross-zone disaster recovery

You can create vSwitches to divide a VPC into one or more subnets. vSwitches within the same VPC can communicate with each other. To implement cross-zone disaster recovery, you can deploy resources across vSwitches in different zones.



Isolate business systems

VPCs are logically isolated from each other. You can use multiple VPCs to isolate business systems in different environments such as production and test environments. To allow business systems deployed in two VPCs to communicate with each other, you can create a peering connection between the VPCs.



Build a hybrid cloud

To expand your on-premises network, you can create a dedicated connection between a VPC and your data center. This allows you to seamlessly migrate the application systems in your data center to the cloud. You do not need to change the access method for the applications.



15.6. Terms

This topic describes the terms of Virtual Private Cloud (VPC).

Term	Description
VPC	A VPC is a private network established on Alibaba Cloud. VPCs are logically isolated from each other. You can create and manage cloud resources in your VPC, such as Elastic Compute Service (ECS), Server Load Balancer (SLB), and ApsaraDB RDS instances.
vSwitch	A vSwitch is a basic network device that connects different cloud resources. When you create a cloud resource in a VPC, you must specify a vSwitch to which the cloud resource is connected.
vRouter	A vRouter is a virtual router that connects all vSwitches in a VPC, and serves as a gateway that connects the VPC to other networks. A vRouter also forwards network traffic based on the route entries in the route table.
Route table	A route table is a list of route entries in a vRouter.
Route entry	Each item in a route table is a route entry. A route entry specifies the next hop address for the network traffic that is destined for a destination CIDR block. Route entries are classified into system route entries and custom route entries.

15.7. Limits

Virtual Private Cloud (VPC) imposes limits on its features. We recommend that you understand the limits before you use the features.

Limits on VPCs and vSwitches

Item	Default limit
Number of VPCs that can be created in each region	10
Number of vSwitches that can be created in each VPC	24
Available CIDR blocks for each VPC	192.168.0.0/16, 172.16.0.0/16, and their subnets.
Number of secondary IPv4 CIDR blocks that can be created in each VPC	1
Number of private IP addresses that can be used by cloud resources in each VPC	- IPv4 addresses: 60,000 - IPv6 addresses: 2,000  Note - If an ECS instance has only one private IP address, the ECS instance uses only one network address. - If an ECS instance is associated with multiple ENIs, or multiple IP addresses are assigned to an ENI, the number of network addresses used by the ECS instance equals the total number of the IP addresses assigned to the ENIs that are associated with the ECS instance.

Limits on vRouters and route tables

Item	Default limit
Number of vRouters that can be created in each VPC	1
Number of custom route entries that can be created in each route table	48
Number of tags that can be added to each route table	20

Limits on network access control lists (ACLs)

Item	Default limit
Number of network ACLs that can be created in each VPC	200
Number of network ACLs that can be associated with each vSwitch	1

Item	Default limit
Number of rules that can be added to a network ACL	• Inbound rules: 20 • Outbound rules: 20

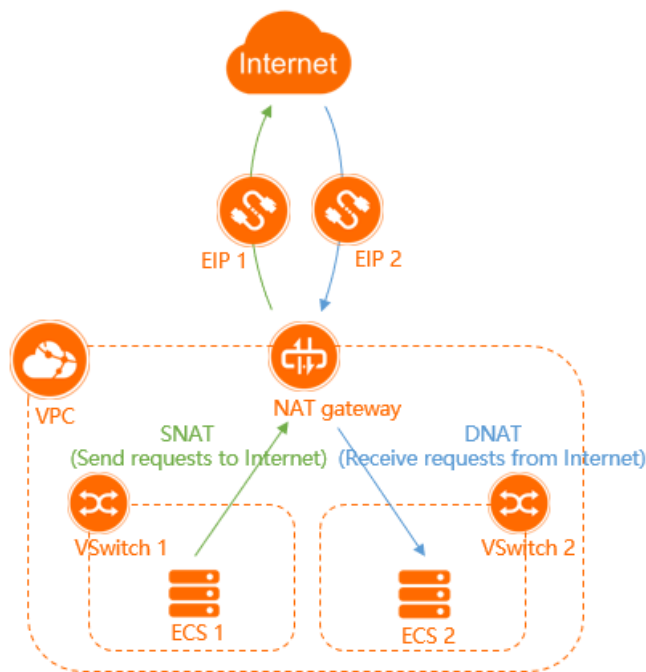
Limits on high-availability virtual IP addresses (HAVIPs)

Item	Default limit
Number of HAVIPs that can be created in each VPC	5
Number of HAVIPs that can be associated with each ECS instance	5
Number of HAVIPs that can be associated with each ENI	5
Number of ECS instances that can be associated with each HAVIP	2
Number of ENIs that can be associated with each HAVIP	2
Number of route entries that point to an HAVIP in each VPC	5
Whether HAVIPs support broadcasting or multicasting	Not supported  Note HAVIPs support only unicasting. To implement high availability through third-party software such as keepalived, you must modify the configuration file to change the communication method to unicasting.

16.NAT Gateway

16.1. What is NAT Gateway?

NAT gateways are enterprise-class gateways that provide the Source Network Address Translation (SNAT) and Destination Network Address Translation (DNAT) features. Each NAT gateway provides a throughput capacity of up to 100 Gbit/s. NAT gateways also support cross-zone disaster recovery.



Features

You must associate public IP addresses with NAT gateways so that the NAT gateways can function as expected. After you create a NAT gateway, you can associate elastic IP addresses (EIPs) with the NAT gateway.

NAT gateways provide the SNAT and DNAT features.

Feature	Description
SNAT	SNAT allows ECS instances that are deployed in a VPC to access the Internet when no public IP address is associated with these ECS instances.
DNAT	DNAT maps the EIPs of a NAT gateway to ECS instances. This way, the ECS instances can receive requests from the Internet.

16.2. Features

NAT gateways support the Source Network Address Translation (SNAT) and Destination Network Address Translation (DNAT) features.

SNAT

NAT gateways support the SNAT feature. SNAT enables Elastic Compute Service (ECS) instances in a virtual private cloud (VPC) to access the Internet when the ECS instances are not assigned public IP addresses.

You can also use the SNAT feature of NAT gateways to protect ECS instances. ECS instances in a VPC can establish connections with the Internet only when the ECS instances access the Internet. External users cannot access ECS instances in the VPC from the Internet. After you use SNAT, the ports that the ECS instances in a VPC use to access the Internet are invisible to external users. This protects the ECS instances in a VPC against external attacks.

DNAT

NAT gateways support the DNAT feature. DNAT maps elastic IP addresses (EIPs) to ECS instances in a VPC. This way, the ECS instances can receive requests that are sent over the Internet.

16.3. Benefits

NAT Gateway features easy configuration, high performance, high availability, and flexible specification adjustment.

Easy configuration

A NAT gateway is an enterprise-grade Internet gateway that controls the traffic flowing to and from a VPC and provides SNAT and DNAT. NAT gateways are reliable, flexible, and easy-to-use, saving you the trouble of building an Internet gateway yourself.

High performance

A NAT gateway is a virtual network device developed based on the software-defined networking (SDN) technology and the distributed gateway design of Alibaba Cloud. NAT Gateway supports a forwarding capability of up to 10 Gbit/s, meeting the requirements of large-scale Internet applications.

High availability

NAT Gateway supports cross-zone deployment, which guarantees high availability and ensures service continuity during a zone failure.

Flexible specification adjustment

You can change the specification of your NAT gateway, or the number and specifications of the EIPs associated with the NAT gateway at any time to provide dynamic support for your services.

16.4. Scenarios

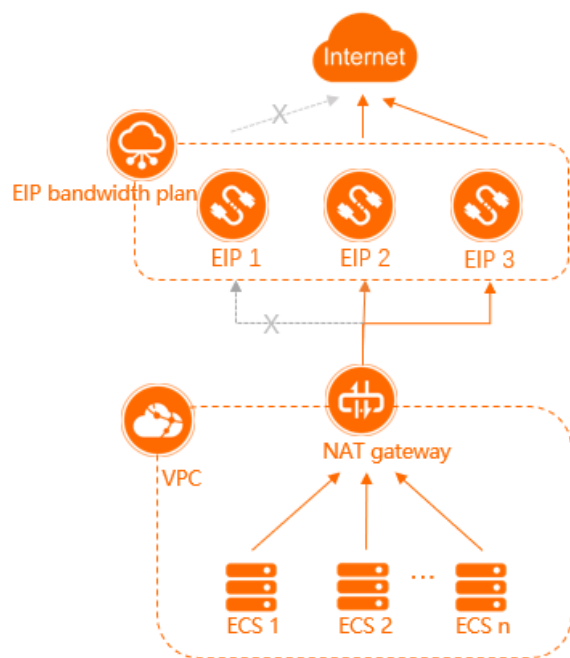
You can use NAT gateways to enable Elastic Compute Service (ECS) instances in virtual private clouds (VPCs) to access the Internet and receive requests from the Internet.

Create a SNAT gateway to enable ECS instances to access the Internet

You can create a NAT gateway for a VPC, associate an elastic IP address (EIP) with the NAT gateway, and then create a Source Network Address Translation (SNAT) entry on the NAT gateway. This way, the ECS instances in the VPC can access the Internet by sharing the EIP. This saves public IP resources.

You can also associate multiple EIPs with the NAT gateway. When an ECS instance needs to access the Internet, it randomly selects an EIP from the SNAT IP address pool. If one of the EIPs is under attack, ECS instance can randomly select another EIP from the SNAT IP address pool to access the Internet. This ensures high availability of your workloads. We recommend that you associate multiple EIPs with a NAT gateway to avoid service interruption caused by EIP failures.

Note Before you associate multiple EIPs with a NAT gateway, make sure that you have added these EIPs to the same EIP bandwidth plan.

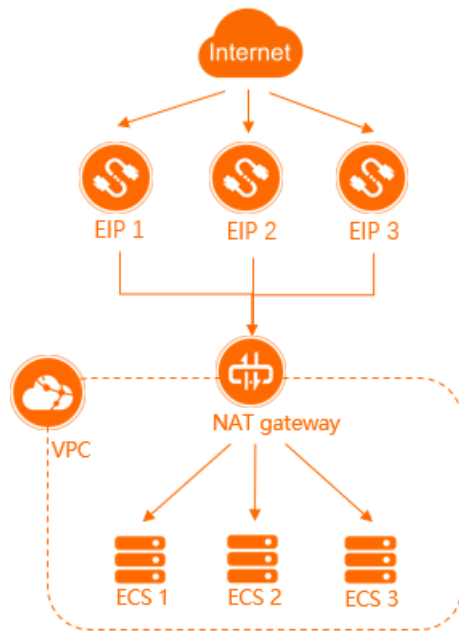


Create a DNAT gateway to enable ECS instances to receive requests from the Internet

You can create a NAT gateway for a VPC, associate EIPs with the NAT gateway, and then create a Destination Network Address Translation (DNAT) entry on the NAT gateway. This way, ECS instances in the VPC can receive requests from the Internet through port mapping or IP mapping.

Note Port mapping and IP mapping are used for the following purposes:

- **Port mapping:** A NAT gateway forwards requests destined for an EIP to the specified ECS instance. Requests are forwarded based on the specified source and destination ports and the specified protocols used by both ports.
- **IP mapping:** A NAT gateway forwards all requests destined for an EIP to the specified ECS instance.



16.5. Terms

The following table lists the terms used in NAT Gateway and their descriptions.

Term	Description
NAT gateways	A NAT gateway is an enterprise-grade Internet gateway that controls the traffic flowing to and from a VPC. NAT Gateway provides SNAT and DNAT features, a maximum forwarding capacity of 10 Gbit/s, and support for cross-zone disaster recovery.
DNAT tables	A DNAT table is a configuration table of a NAT gateway and is used to configure DNAT. DNAT allows you to map a public IP address of a NAT gateway to an ECS instance through IP or port mapping.
SNAT tables	An SNAT table is a configuration table of a NAT gateway and is used to configure SNAT. You can configure an SNAT entry for a VSwitch or for a specific ECS instance. - Configure an SNAT entry for a VSwitch: All ECS instances in the VSwitch use the specified public IP addresses to access the Internet. - Configure an SNAT entry for an ECS instance: The ECS instance uses the specified public IP addresses to access the Internet.
EIPs	An EIP is a public IP address that you can purchase and own independently. You can associate an EIP with an ECS instance deployed in a VPC, an internal SLB instance deployed in a VPC, or a secondary elastic network interface (ENI) attached to a VPC. You can also associate an EIP with a NAT gateway or a High-availability Virtual IP Address (HAVIP).

16.6. Limits

This topic describes the limits on NAT Gateway and how to apply for a quota increase.

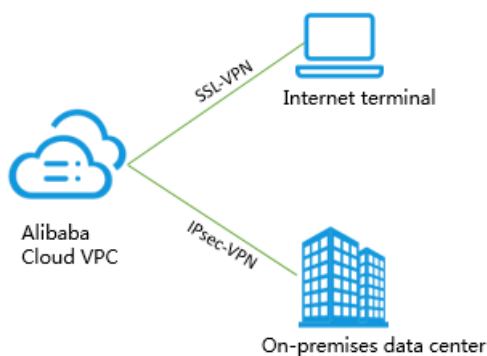
Item	Limit
The maximum number of NAT gateways that can be created for a VPC	1
Use an EIP for both SNAT and DNAT	Not supported
The maximum number of DNAT entries that can be added to a NAT gateway	100
The maximum number of SNAT entries that can be added to a NAT gateway	40
The maximum number of EIPs that can be specified in a SNAT entry	64
Create a NAT gateway for a VPC that contains a custom route entry whose destination CIDR block is 0.0.0.0/0	Not supported  Note Before you can create a NAT gateway for the VPC, you must delete the custom route entry whose destination CIDR block is 0.0.0.0/0.
The bandwidth of a vSwitch is limited by the bandwidth limit of the EIPs in the SNAT entry that is created for the vSwitch	Yes
The number of EIPs that can be associated with a NAT gateway	20
ECS instances in a VPC may access the same destination IP address and port on the Internet by using a NAT gateway. If static public IP addresses or EIPs are not associated with ECS instances, the maximum number of concurrent connections that the NAT gateway supports is based on the number of EIPs in the SNAT entry. - If only one EIP is specified, the maximum number of concurrent connections that the NAT gateway supports is 55,000. - If multiple EIPs are specified, the maximum number of concurrent connections that the NAT gateway supports is $N \times 55,000$. In this formula, N represents the number of the EIPs.	

17.VPN Gateway

17.1. What is VPN Gateway?

VPN Gateway is an Internet-based service that allows you to connect enterprise data centers, office networks, or Internet-facing terminals to Alibaba Cloud Virtual Private Cloud (VPC) networks through secure and reliable connections. VPN Gateway supports both IPsec-VPN connections and SSL-VPN connections.

Note The Alibaba Cloud VPN Gateway service complies with the local regulations and policies. VPN Gateway does not provide Internet access services.



Features

VPN Gateway supports the following features:

- IPsec-VPN

Route-based IPsec-VPN allows you to route network traffic in multiple ways, and also facilitates the configuration and maintenance of VPN policies.

You can use IPsec-VPN to connect an on-premises data center to a VPC network or connect two VPC networks. IPsec-VPN supports the IKEv1 and IKEv2 protocols. Any devices that support these two protocols can connect to Alibaba Cloud VPN Gateway, such as devices manufactured by Huawei, H3C, Hillstone, Sangfor, Cisco ASA, Juniper, SonicWall, Nokia, IBM, and Ixia.

- SSL-VPN

SSL-VPN is implemented based on the OpenVPN framework. You can create an SSL-VPN connection to connect a remote client to applications and services deployed in a VPC network. After you deploy your applications or services, you only need to import the certificate to the client to initiate a connection.

Benefits

VPN Gateway offers the following benefits:

- High security: You can use the IKE and IPsec protocols to encrypt data for secure and reliable data transmission.
- High availability: VPN Gateway adopts the hot-standby architecture to achieve failover within a few seconds, session persistence, and zero service downtime.

- Cost-effectiveness: The encrypted Internet connections provided by VPN Gateway are more cost-effective than leased lines.
- Ease of use: VPN Gateway is a ready-to-use service. VPN gateways start to work immediately after they are deployed.

17.2. Scenarios

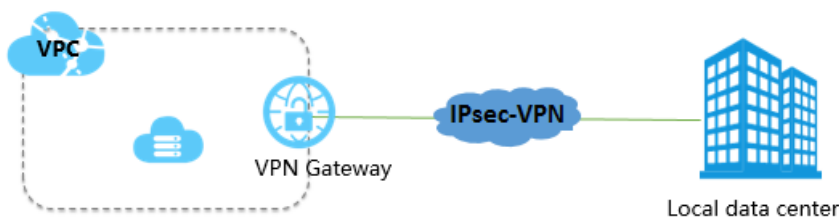
VPN Gateway is an Internet-based service that allows you to connect enterprise data centers, office networks, or Internet-facing terminals to Alibaba Cloud Virtual Private Cloud (VPC) networks through encrypted connections. VPN Gateway can be configured based on your requirements and applied in multiple scenarios.

Connect a VPC network to an on-premises data center

You can use IPsec-VPN to connect an on-premises data center to a VPC network and build a hybrid cloud.

Route-based IPsec-VPN allows you to route network traffic in multiple ways, and also facilitates the configuration and maintenance of VPN policies. For more information, see [Connect a data center to a VPC](#).

Note Before you create an IPsec-VPN connection between an on-premises data center and a VPC network, make sure that the IP address of the on-premises data center is different from that of the VPC network. In addition, you must set a static public IP address for your VPN gateway.

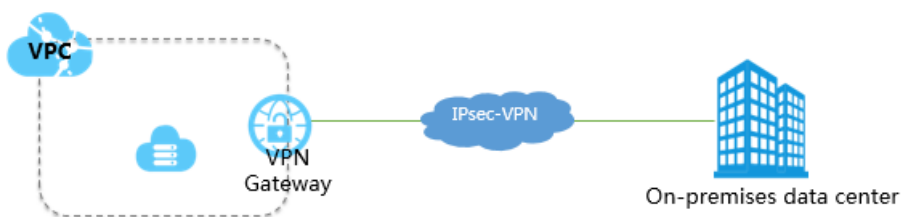


Connect two VPC networks

You can use IPsec-VPN to connect two VPC networks. This way, cloud resources can be shared across these VPC networks.

Route-based IPsec-VPN allows you to route network traffic in multiple ways, and also facilitates the configuration and maintenance of VPN policies. For more information, see [Establish IPsec-VPN connections between two VPCs](#).

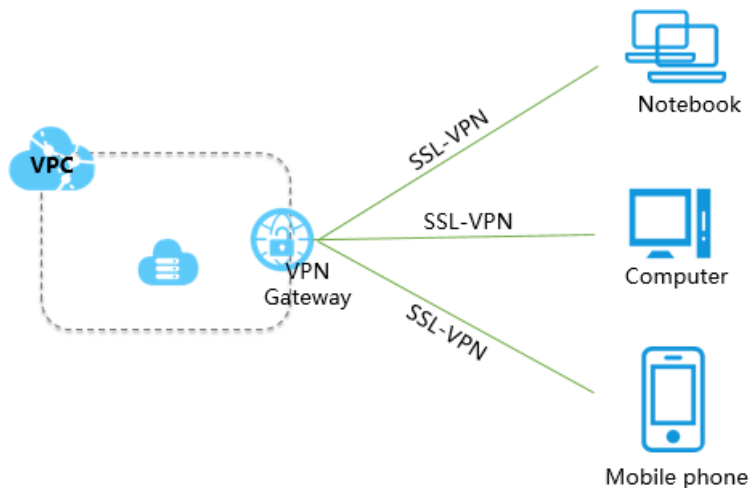
Note The CIDR blocks of the two VPC networks must not overlap with each other.



Connect a VPC network to a mobile client

If you require office automation, you can use SSL-VPN to connect a mobile client to a VPC network. This way, the mobile client can access cloud resources deployed in the VPC network anytime and anywhere. You can initiate an SSL-VPN connection from clients that run Windows, Linux, macOS, iOS, and Android. For more information, see [Create a remote connection from a Linux client](#), [Create a remote connection from a Windows client](#) and [Create a remote connection from a macOS client](#).

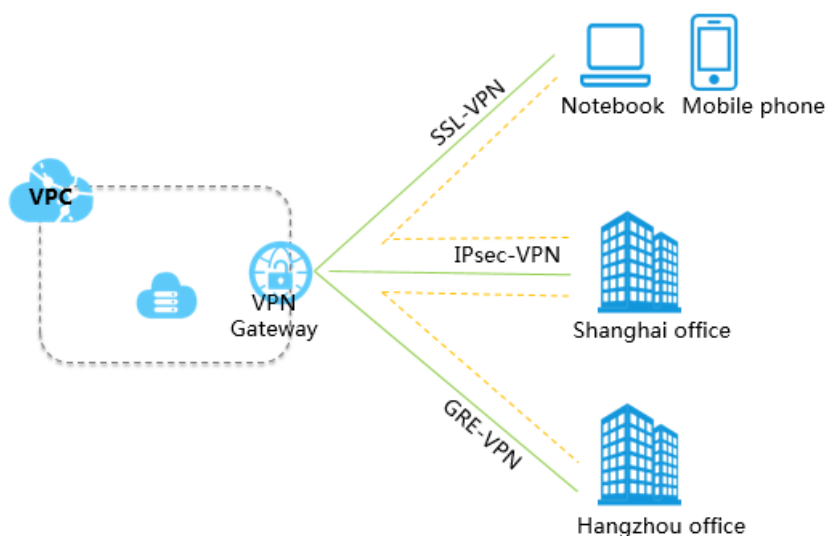
Note The CIDR block assigned to the client must not overlap with that of the VSwitch in the VPC network.



Use IPsec-VPN and SSL-VPN together

You can use IPsec-VPN and SSL-VPN connections together to expand your network topology. After the connections are established, the client can access the applications deployed in the connected VPC network, and can also access the applications deployed in the connected office networks.

Note The private CIDR blocks to be interfaced must not overlap with each other.



17.3. Limits

Before you use VPN Gateway, note the following limits.

VPN gateways

Item	Limit
Maximum number of VPN gateways for each account	30  Note The maximum number of VPN gateways that can be created is determined by the account, regardless of the region where the VPN gateways are deployed or the connected VPC networks. For example, for each account: • If there is only one VPC network in a region, you can create up to 30 VPN gateways for the VPC network in the region. • If there are multiple regions or VPC networks, you can create a total number of 30 VPN gateways for the VPC networks in all regions.
Maximum number of policy-based routes for a VPN gateway	20
Maximum number of destination-based routes that can be created for a VPN gateway	20

Customer gateways

Item	Limit
Maximum number of customer gateways that you can create in a region	100

IPsec-VPN connections

Item	Limit
Maximum number of IPsec-VPN connections for a VPN gateway	10
Maximum number of local CIDR blocks that can be added to an IPsec-VPN connection	5

Item	Limit
Maximum number of remote CIDR blocks that can be added to an IPsec-VPN connection	5

SSL-VPN connections

Item	Limit
Maximum number of SSL client certificates that an account can reserve	50
Maximum number of SSL servers that can be associated with a VPN gateway	1
Maximum number of local CIDR blocks that can be added to an SSL server	5
Maximum number of remote CIDR blocks that can be added to an SSL server	1
Ports that are not supported by SSL servers	22, 2222, 22222, 9000, 9001, 9002, 7505, 80, 443, 53, 68, 123, 4510, 4560, 500, and 4500
The validity period of an SSL client certificate	Three years

18. Elastic IP Address

18.1. What is Elastic IP Address?

This topic provides an overview of Elastic IP Address. An elastic IP address (EIP) is a public IP address that you can purchase and use as an independent resource. You can associate an EIP with an Elastic Compute Service (ECS) instance, an internal-facing Server Load Balancer (SLB) instance, or a secondary elastic network interface (ENI) deployed in a virtual private cloud (VPC). You can also associate an EIP with a NAT gateway, or a high-availability virtual IP address (HAVIP).

An EIP is a NAT IP address provisioned in the Internet-facing gateway of Alibaba Cloud and is mapped to the associated cloud resource by using NAT. After an EIP is associated with a cloud resource, the cloud resource can use the EIP to communicate with the Internet.

Differences between an EIP and the static public IP address of an ECS instance

The following table describes the differences between an EIP and the static public IP address of an ECS instance.

Item	EIP	Static public IP address
Supported network	VPC	VPC
Used as an independent resource	Supported	Not supported
Associated with and disassociated from an ECS instance at any time	Supported	Not supported
Displayed in the ENI information of the operating system of the associated ECS instance	Not displayed.	Not displayed

Benefits

EIPs have the following benefits:

- Purchase and use as independent resources

You can purchase and use an EIP as an independent resource. EIPs are not bundled with other computing or storage resources.

- Associate with resources at any time

You can associate an EIP with a cloud resource as needed. You can also disassociate and release the EIP at any time.

- Modify bandwidth limits on demand

You can modify the bandwidth limit of an EIP at any time to meet your business requirements. The modification immediately takes effect.

18.2. Limits

This topic describes the limits of Elastic IP Address (EIP).

General limits

General limits of EIPs:

- You can associate an EIP with only one cloud resource. The EIP that you want to associate must be in the Available state. After you associate an EIP with a cloud resource, the EIP immediately takes effect.
- If an EIP is locked for security purposes, you cannot release the EIP, associate the EIP with a cloud resource, or disassociate the EIP from a cloud resource.

Limits on associating an EIP with an ECS instance

Before you associate an EIP with an Elastic Compute Service (ECS) instance, take note of the following limits:

- The ECS instance must be deployed in a virtual private cloud (VPC).
- The ECS instance and the EIP must belong to the same region.
- The ECS instance must be in the Running or Stopped state.
- The ECS instance is not assigned a static public IP address. In addition, the ECS instance is not associated with another EIP.
- Each ECS instance can be associated with only one EIP. If you want an ECS instance to use multiple EIPs, you can attach secondary ENIs that are associated with EIPs to the ECS instance. You can use one of the following methods to perform this task:
 - Associate multiple EIPs with a secondary ENI in **NAT mode**. In this mode, each EIP is associated with a secondary private IP address of the secondary ENI. Then, associate the secondary ENI with the ECS instance.

Limits on associating an EIP with a NAT gateway

Before you associate an EIP with a NAT gateway, take note of the following limits:

- The NAT gateway and the EIP must belong to the same region.

Limits on associating an EIP with an SLB instance

Before you associate an EIP with a Server Load Balancer (SLB) instance, take note of the following limits:

- The SLB instance must be deployed in a VPC.
- The SLB instance and the EIP must belong to the same region.
- You can associate only one EIP with each SLB instance.

Limits on associating an EIP with an HAVIP

Before you associate an EIP with a high-availability virtual IP address (HAVIP), take note of the following limits:

- The HAVIP and the EIP must belong to the same region.
- The HAVIP must be in the Available or Allocated state.
- You can associate only one EIP with each HAVIP.

Limits on associating an EIP with a secondary ENI

Before you associate an EIP with a secondary ENI, take note of the following limits:

- The secondary ENI must be deployed in a VPC.
- The secondary ENI and the EIP must belong to the same region.
- You can associate EIPs with a secondary ENI only in NAT mode. The number of EIPs that you can associate with a secondary ENI in NAT mode depends on the number of private IP addresses of the secondary ENI.

Limits on purchasing EIPs of different Border Gateway Protocol (BGP) lines

Starting 00:00 (UTC+8) on December 22, 2020, you cannot purchase EIPs of **BGP (Multi-ISP) International** in the China (Hong Kong) region. EIPs of **BGP (Multi-ISP) International** that are purchased before 00:00 (UTC+8) on December 22, 2020 are not affected.

19. Log Service

19.1. What is Log Service?

Log Service (SLS) is a one-stop logging service developed by Alibaba Cloud that is widely used by Alibaba Group in big data scenarios. You can use Log Service to collect, query, and consume log data without the need to invest in in-house data collection and processing systems. This enables you to focus on your business, improving business efficiency and helping your business to expand.

Log Service provides the following features:

- **Log collection:** Log Service allows you to collect events, binary logs, and text logs in real time through multiple methods, such as Logtail and JavaScript.
- **Query and analysis:** Log Service allows you to query and analyze the collected log data and view analysis results on charts and dashboards.
- **Status alert:** Log Service can automatically run query statements at regular intervals after you create an alert task. If the query results meet the conditions of the alert task, Log Service sends an alert to the specified recipients in real time.
- **Real-time consumption:** Log Service provides real-time consumption interfaces through which log consumers can consume log data.

19.2. Benefits

This topic describes the benefits of Log Service.

Fully managed service

- Log Service is easy to access and use.
- LogHub provides all features of Kafka, outputs monitoring and alert data, and supports auto scaling (by PBs per day).
- LogSearch/Analytics allows you to query log data, view log data on dashboards, and configure alerts.
- Log Service provides more than 30 access methods to seamlessly connect with open-source software, such as Storm and Spark Streaming.

Comprehensive ecosystem

- LogHub supports more than 30 types of log data sources, such as embedded devices, web pages, servers, and programs. LogHub can connect with consumption systems, such as Storm and Spark Streaming.
- LogSearch/Analytics provides complete query and analysis syntax and supports connection with Grafana based on the SQL-92 and JDBC protocols.

Real-time response

- **LogHub:** Data can be consumed immediately after being written to Log Service. Logtail acts as an agent to collect and deliver data in real time.
- **LogSearch/Analytics:** Data can be queried and analyzed immediately after being written to Log Service. In scenarios where multiple query and analysis conditions are specified, data can be queried and analyzed in seconds.

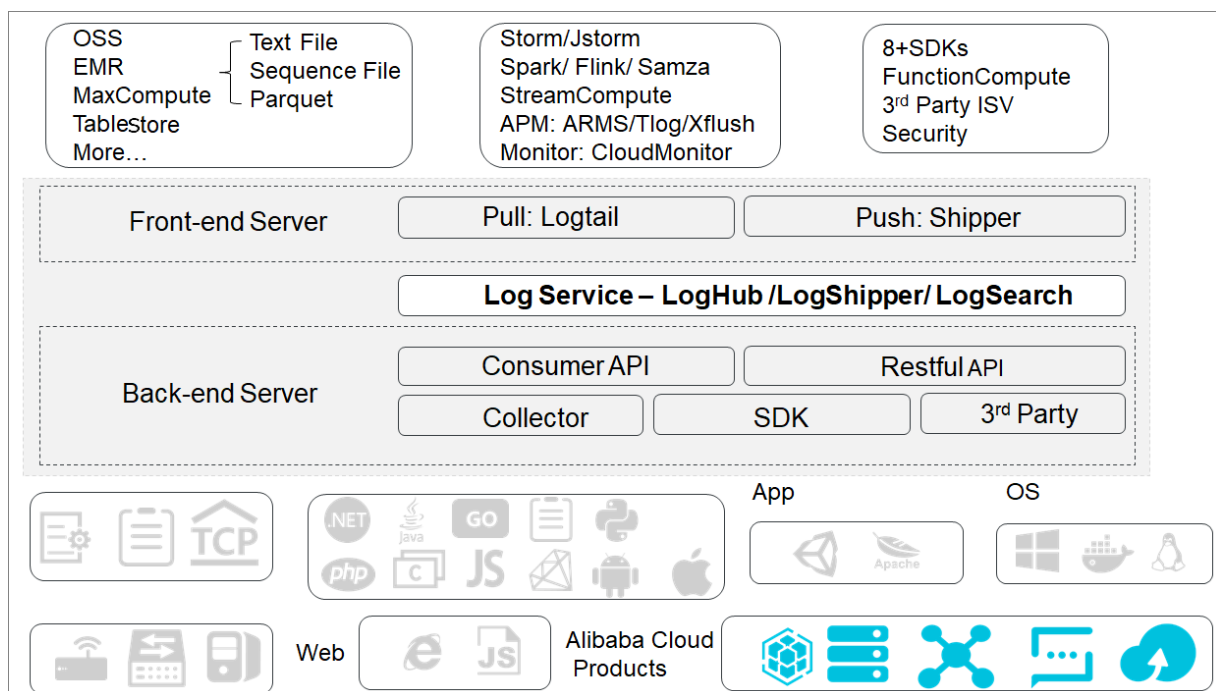
Scalability

Log Service provides auto scaling capabilities for petabytes of data.

19.3. Architecture

This topic describes the architecture of Log Service.

The following figure shows the architecture of Log Service.



Logtail

Logtail is an agent that is used to collect logs. It has the following features:

- Non-intrusive file-based log collection
 - Logtail only reads log files.
 - Log collection is non-intrusive.
- High security and reliability
 - Logtail can rotate logs without data loss.
 - Logtail supports local caching.
 - Logtail re-attempts to collect logs if network exceptions occur.
- Ease management
 - Logtail can be accessed by using a web client.
 - Logtail can be configured in the console.
- Comprehensive self-protection
 - Logtail monitors CPU and memory usage of its processes in real time.
 - Logtail allows you to set an upper limit on the resource usage of its processes.

Frontend servers

Frontend servers are built on LVS and NGINX to provide the following features:

- Data transfer over HTTP and REST protocols
- Horizontal scaling
 - The processing capabilities can be increased when traffic increases.
 - Frontend servers can be added.
- High throughput and low latency
 - Asynchronous processing: If an exception occurs when a single request is sent, other requests are not affected.
 - LZ4 compression: This compression method increases the processing capabilities of individual servers and reduces network bandwidth consumption.

Backend servers

The backend service is a distributed process that is deployed on multiple servers. The service can store, index, and query logs in a Logstore in real time. The backend service has the following features:

- High data security
 - Each log is saved to three copies and stored on different servers.
 - If the disk is damaged or the server fails, data is automatically copied and restored.
- Stable services
 - If the process does not respond or the server fails, data in Logstores is automatically migrated.
 - Automatic load balancing ensures that traffic is evenly distributed among different servers.
 - Strict resource quota limits prevent unexpected operations of a single user from affecting other users.
- Horizontal scaling
 - A shard is the basic unit for horizontal scaling.
 - You can add shards to increase the throughput based on your business requirements.

19.4. Features

19.4.1. Core features

This topic describes the two core features of Log Service: real-time log collection and consumption (LogHub) and real-time log search and analysis (Log Search/Analytics).

LogHub

LogHub allows you to collect logs without data loss by using various methods. These methods include clients, websites, protocols, SDKs, and API operations (for mobile apps and games). You can also consume logs by using SDKs, Storm Spout, and Spark Client. LogHub supports real-time log collection and consumption in multiple formats. You can use LogHub to streamline the collection and consumption of logs across multiple devices and sources.

Features:

- LogHub collects real-time log data from Elastic Compute Service (ECS) instances, containers, mobile terminals, open-source software, and JavaScript. The log data can be metrics, events, binary logs, and text logs.

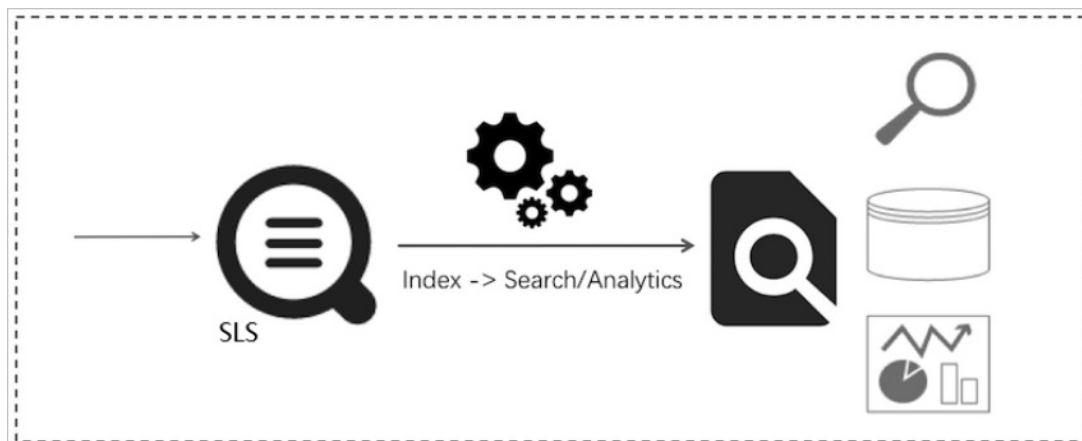
- A real-time consumption interface is provided to connect Log Service to real-time computing engines and services.

Log Search/Analytics

You can use Log Service to index, query, and analyze collected log data in real time. Log Service can generate dynamic reports based on the query and analysis results. It can also generate visualized reports of log data in multiple scenarios.

- Query: You can query data by using keywords, fuzzy match, and contextual query. You can also query data within a specified range.
- Statistics: You can use various query methods, such as SQL aggregate functions.
- Visualization: You can view log analysis on dashboards and export the results as reports.
- Interconnection: You can connect Log Service to Grafana based on the Java Database Connectivity (JDBC) and SQL-92 protocols.

Log search/analytics



19.4.2. Other features

19.4.2.1. Logs

Logs are records of changes that occur in a system. The operations on specific objects and the relevant results are recorded in logs in chronological order.

Logs in Log Service

Log Service supports different types of logs, such as log files, events, binary logs, and metrics. Each log file consists of one or more log entries. Each log entry describes a single system event and is the smallest unit of data that can be processed in Log Service.

Log Service uses a semi-structured data model to define logs. This model consists of the topic, time, content, and source fields.

Log Service has different format requirements on different log fields, as described in the following table.

Field	Description	Format

Field	Description	Format
Topic	The user-defined field in a log. This field can be used to mark a group of logs. For example, access logs can be marked based on sites.	The value of this field can be a string up to 128 bytes in length, including an empty string. The default value of this field is an empty string.
Time	The time when a log was generated. This field is a reserved field. The value of this field is directly extracted from the time in the log.	This value is a Unix timestamp representing the number of seconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC.
Content	The specific content of a log. The content consists of one or more content items. Each item is a key-value pair.	The key is a UTF-8 encoded string up to 128 bytes in length. It can contain letters, digits, and underscores (_). The key cannot start with a digit and cannot contain the following keywords: <code>_time</code> , <code>source</code> , <code>topic</code> , <code>partition</code> , <code>time</code> , <code>extract</code> , <code>others</code> , and <code>__extract_others__</code> . The value can be a string up to 1024 × 1024 bytes in length.
Source	The source of a log. For example, the source can be the IP address of the server where the log was generated.	The value of this field can be a string up to 128 bytes in length. The default value of this field is an empty string.

Logs are used in various formats in different scenarios. The following example shows how to map a raw NGINX access log to the log data model of Log Service. In this example, the IP address of your NGINX server is `10.249.201.117`. A raw log generated on this server is as follows:

```
10.1.168.193 - - [01/Mar/2012:16:12:07 +0800] "GET /Send? AccessKeyId=8225105404 HTTP/1.1" 200 5 "-" "Mozilla/5.0 (X11; Linux i686 on x86_64; rv:10.0.2) Gecko/20100101 Firefox/10.0.2"
```

The following table describes how to map this raw log to the log data model of Log Service.

Field	Field value	Description
Topic	""	An empty string is used.
Time	1330589527	The exact time when the log was generated is used. The value indicates the number of seconds that have elapsed since the epoch time January 1, 1970, 00:00:00 UTC. The value is converted from the timestamp in the raw log.

Field	Field value	Description
Content	Key-value pair	The specific content of the log is used.
Source	"10.249.201.117"	The IP address of the server is used as the log source.

You can decide how to extract the content of a raw log to create key-value pairs. The following table lists some key-value pairs.

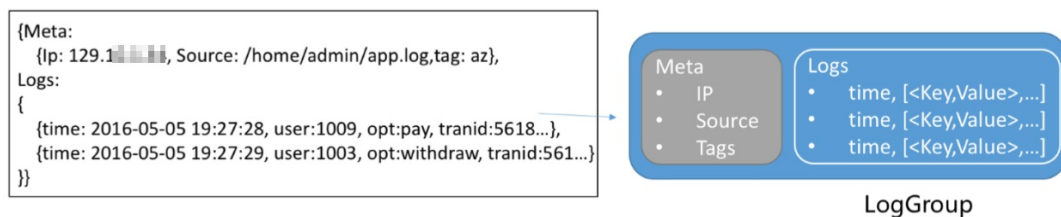
Key	Value
ip	10.1.168.193
method	GET
status	200
length	5
ref_url	N/A
browser	Mozilla/5.0 (X11; Linux i686 on x86_64; rv:10.0.2) Gecko/20100101 Firefox/10.0.2

Log groups

A log group is a collection of logs and is the basic unit for read and write operations.

The maximum capacity of a log group is 4,096 logs or 10 MB of logs.

Log group



19.4.2.2. Project

This topic introduces the concept of project. It also describes the features of a project.

A project in Log Service is the resource management unit that is used to separate and manage different resources. You can use a project to manage all the logs and related log sources of an application. You can also use a project to manage Logstores and Logtail configuration files that are used to collect logs. A project serves as an endpoint that allows authorized access to the resources of Log Service.

Projects provide the following features:

- Projects allow you to manage different Logstores. Logs that you collect and store in Log Service are generated from different projects, services, and environments. You can specify different projects for

these logs to facilitate data consumption, exporting, and indexing. You can also grant permissions on these projects to different users.

- Projects serve as endpoints that allow authorized access to the resources of Log Service. Log Service allocates an exclusive endpoint to each project. You can use the endpoint of a project to read, write, and manage log data.

19.4.2.3. Logstore

This topic introduces the concept of Logstore. It also describes the features of a Logstore.

A Logstore in Log Service is the unit that is used to collect, store, and query data. Each Logstore belongs to only one project. However, you can create multiple Logstores for a single project. In most cases, a Logstore is created for each type of log in an application. For example, you have a gaming application called big-game, and three types of logs are stored on the server: operation_log, application_log, and access_log. You can create a project named big-game, and then create three Logstores in this project to collect, store, and query these logs.

You must specify a Logstore when you write or query logs. If you transfer log data to MaxCompute for offline analysis, the logs in each Logstore are transferred to a MaxCompute Table.

Logstores provide the following features:

- Logs can be written to Logstores in real time.
- Logs stored in Logstores can be consumed in real time.
- Indexes can be created for Logstores to support real-time log queries.

19.4.2.4. Shard

This topic describes the range, read/write capacities, and status of the shards of a Logstore.

Log data is stored on a shard of a Logstore where read/write operations are performed. A Logstore consists of multiple shards. Each shard has an MD5 hash key range. Each range is left-closed and right-open and does not overlap with the ranges of other shards. The entire range of MD5 hash key values consists of all these different ranges.

Range

You must specify the number of shards when you create a Logstore. The entire range of MD5 hash key values is evenly divided by Log Service based on the specified number of shards. The MD5 hash key ranges of the shards must be within the following range: [00000000000000000000000000000000, ffffffffffffffffffffffffffffffffff).

The range of each shard is a left-closed and right-open interval. The range consists of the following keys:

- BeginKey: indicates the start of a shard. This value is included in the range.
- EndKey: indicates the end of a shard. This value is excluded from the range.

You can use the hash key values of the ranges to write logs to specified shards. You can also use the values to identify shards that you want to split or merge. When you read data from a shard, you must specify the shard. When you 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 range includes the value of the specified hash key.

For example, a Logstore has four shards and the MD5 hash value range of the Logstore is [00, FF). The following table describes the ranges of the shards.

Shard ID	MD5 hash key range
Shard0	[00,40)
Shard1	[40,80)
Shard2	[80,C0)
Shard3	[C0,FF)

If you set the MD5 hash key value to 5F when you write logs in the hash key mode, the log data is written to Shard1 because the range of Shard1 contains the MD5 hash key value 5F. If you set the MD5 hash key value to 8C, the log data is written to Shard2 because the range of Shard2 contains the MD5 hash key value 8C.

Read/write capacities

Each shard has limits on read/write capacities. We recommend that you adjust the number of shards based on the actual data traffic. If the data traffic exceeds the read/write capacities of a shard, you can split the shard into two shards to increase the total read/write capacities. If the data traffic is much less than the read/write capacities of a shard, you can merge the shard with the adjacent shard to reduce the total read/write capacities and save costs.

Note

- If a 403 or 500 error code is repeatedly returned when you use API operations to write logs to a Logstore, you can check the Log Service monitoring metrics. You can then determine whether to increase the number of shards.
- If the data traffic exceeds the read/write capacities of your shards, Log Service attempts to provide the best services possible. However, service quality cannot be guaranteed.

The status of the shards

A shard can be in one of the following states:

- readwrite: The shard supports read/write operations.
- readonly: The shard supports only read operations.

When you create a shard, the shard is in the readwrite state. After you split or merge shards, the original shards are in the readonly state and the new shards are in the readwrite state. The status of a shard does not affect its read performance. You can write data to a shard in the readwrite state, but you cannot write data to a shard in the readonly state.

When you split a shard, you must specify the ID of a shard that is in the readwrite state and an MD5 hash key. The value of the MD5 hash key must be greater than the value of the BeginKey and less than the value of the EndKey of a shard. After a shard is split, two shards are added. The status of the original shard changes from readwrite to readonly. Data can still be read from the shard, but cannot be written to the shard. The two new shards are in the readwrite state. They are placed next to the original shard. The total MD5 hash key ranges of the two shards cover the range of the original shard.

When you merge shards, you must specify a shard that is in the readwrite state. In addition, the shard cannot be the last shard that is in the readwrite state. After you specify a shard that is in the readwrite state, Log Service finds the adjacent shard and merges these two shards into a single shard. After you merge the shards, the status of the original shards change from readwrite to readonly. Data can still be read from the shards, but cannot be written to the shards. A shard in the readwrite state is generated. The MD5 hash key range of the shard covers the total range of the original shards.

19.4.2.5. Log topic

This topic introduces the concept of log topic.

A log topic is used to classify logs in a Logstore. You can specify a topic for logs that are written to Log Service. You can also specify a topic when you query logs. For example, you can use your user ID as the log topic when you write logs to Log Service. In this way, you can view only your own logs based on the log topic when you query the logs. If you do not need to classify logs in a Logstore, use the same topic for all logs.

 **Note** The default log topic for log data writes and queries is an empty string. If you do not specify a log topic, you can use the default topic to write or query logs.

19.5. Scenarios

This topic describes the application scenarios of Log Service. The scenarios include data collection, real-time computing, data warehousing, offline analysis, product operation and analysis, and O&M.

Data collection and consumption

You can use the LogHub feature of Log Service to collect large amounts of log data in real time. The log data can be metrics, events, binary logs, and text logs.

Benefits:

- **Ease of use:** More than 30 real-time data collection methods are provided. This simplifies the construction of log O&M platform and reduces your O&M workloads.
- **Elastic scalability:** Log Service supports automatic scaling based on traffic and business requirements. This simplifies the handling of traffic spikes that result from business growth.

ETL and stream processing

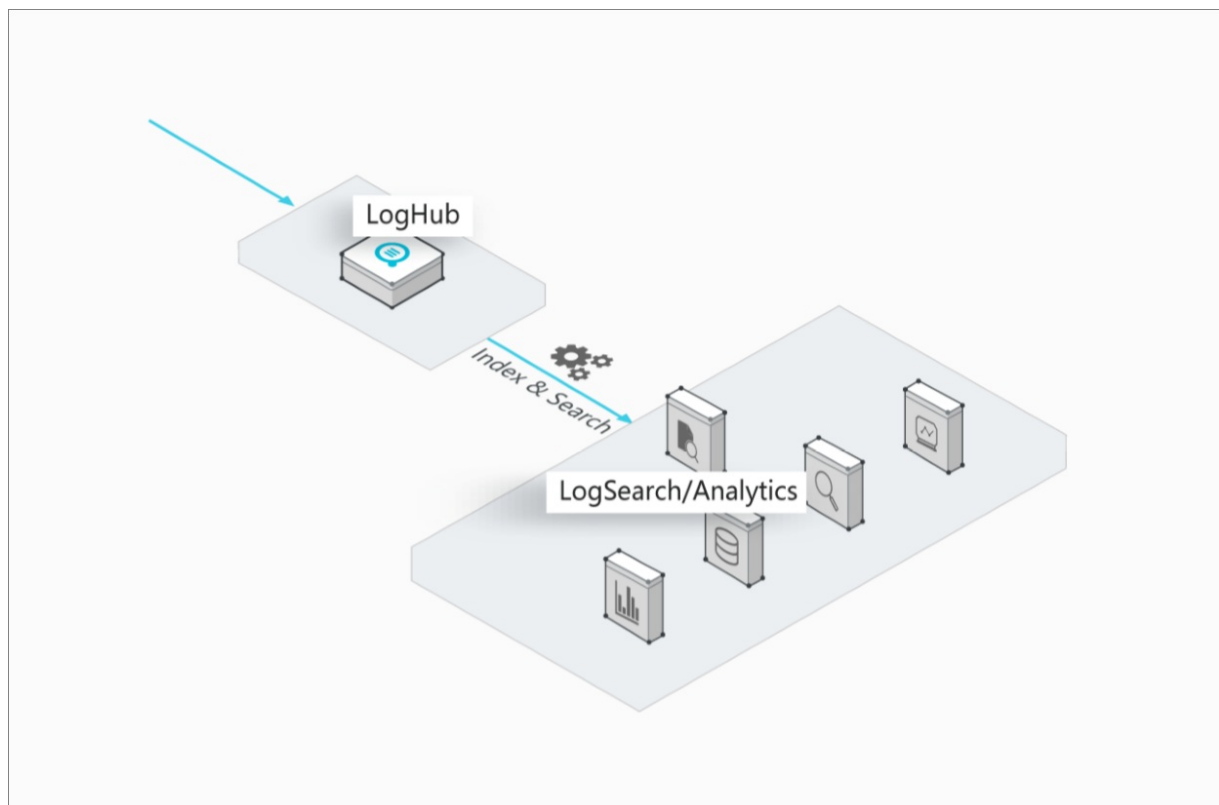
You can connect LogHub to multiple real-time computing engines and services. LogHub can monitor the processing progress and generate alerts based on the monitoring results. You can also use SDKs or API operations to consume logs.

- **Ease of use:** LogHub provides SDKs in multiple programming languages and frameworks. You can connect LogHub to various stream computing engines.
- **Comprehensive features:** LogHub can monitor large amounts of data and generate alerts based on the monitoring results.
- **Elastic scalability:** LogHub supports scaling to handle petabyte of data with zero latency.

Real-time log search and analysis

The log search/analytics feature allows you to index LogHub data in real time and query data by using keywords, fuzzy match, contextual query, and SQL aggregate functions. You can also query data within a specified range.

- Real-time query: Data can be immediately queried after it is written to Log Service.
- High efficiency and low costs: You can index petabytes of data each day. Costs are 85% lower compared with user-defined systems.
- Strong analysis capability: Multiple query methods and SQL aggregate functions are supported. Log Service can also generate visualized reports and alerts based on log data.



19.6. Limits

This topic describes the limits of Log Service.

Limits of resources

Resource	Limit	Note
Project	You can create a maximum of 10 projects in an organization.	If you need to increase the quota, submit a ticket.
Logstore	You can create a maximum of 100 Logstores in a project.	If you need to increase the quota, submit a ticket.

Resource	Limit	Note
Shard	- You can create a maximum of 10 shards in a Logstore. However, you can split the shards to increase the number of shards. - You can create a maximum of 100 shards in a project.	If you need to increase the quota, submit a ticket.
Dashboard	- You can create a maximum of five dashboards for a project. - Each dashboard can contain a maximum of 10 charts.	If you need to increase the quota, submit a ticket.
Saved search	You can create a maximum of 10 saved searches for a project.	If you need to increase the quota, submit a ticket.
Logtail configuration file	You can create a maximum of 100 Logtail configuration files in a project.	If you need to increase the quota, submit a ticket.
Consumer group	You can create a maximum of 10 consumer groups in a project.	If you need to increase the quota, submit a ticket.
Machine group	You can create a maximum of 100 machine groups in a project.	If you need to increase the quota, submit a ticket.
Log retention time	Logs that are collected to the server can be kept for a maximum of 365 days.	If you need to increase the quota, submit a ticket.

19.7. Terms

This topic introduces the basic concepts of Log Service.

Log

Logs are abstract records of changes within a system. The records are ordered by time. The records contain information about operations on specific objects and results of the operations. Log data is stored in different forms such as log files, log events, binary logs, and metric data. Each log file consists of one or more log entries. A log entry is the basic unit of data that can be processed in Log Service. Each log entry describes a single system event.

Log group

A log group is a collection of logs. The groups are the basic units that are used for read and write operations.

Log topic

A log topic is used to classify logs in a Logstore. Topics can be specified when logs are written to Log Service. Topics also serve as filters when logs are queried.

Project

A project in Log Service is the resource management unit that is used to separate and manage different resources. You can use a project to manage all the logs and related log sources of an application. You can also use a project to manage Logstores and Logtail configuration files. A project serves as an endpoint to access the resources of Log Service.

Logstore

A Logstore is the unit used in Log Service for log data collection, storage, and query. Each Logstore belongs to only one project. However, you can create multiple Logstores for a single project.

Shard

A Logstore consists of multiple shards. Each shard has an MD5 hash key range. The range is left-closed and right-open and does not overlap with the ranges of other shards. The entire range of MD5 hash key values consists of all these different ranges.

20. Apsara Stack Security

20.1. What is Apsara Stack Security

Apsara Stack Security is a solution that protects Apsara Stack assets with a full suite of security features, such as network, server, application, data, and security management.

Background information

Traditional security solutions for IT services detect attacks on network perimeters. These solutions use hardware products such as firewalls and intrusion prevention systems (IPSs) to protect networks against attacks.

With the development of cloud computing, an increasing number of enterprises and organizations use cloud computing services instead of traditional IT services. Cloud computing features low costs, on-demand flexible configuration, and high resource utilization. Cloud computing environments do not have definite network perimeters. As a result, traditional security solutions cannot effectively safeguard cloud assets.

With the powerful data analysis capabilities and professional security operations team of Alibaba Cloud, Apsara Stack Security provides integrated security protection services for networks, applications, and servers.

Complete security solution

Apsara Stack Security consists of Apsara Stack Security Standard Edition and optional security services to provide a comprehensive security solution.

Security domain	Service name	Description
Security management	Threat Detection Service (TDS)	Monitors traffic and overall security status to audit and centrally manage assets.
Server security	Server Guard	Protects Elastic Compute Service (ECS) instances against intrusions and malicious code.
	Server Security	Protects physical servers against intrusions.
Application security	Web Application Firewall (WAF)	Protects web applications against attacks and ensures that mobile and PC users can securely access web applications over the Internet.
Network security	Anti-DDoS	Ensures the availability of network links and improves business continuity.
	Cloud Firewall	Allows you to centrally manage access control policies for traffic transferred within your business system (east-west) and between the Internet and your business system (north-south).
Data security	Sensitive Data Discovery and Protection (SDDP)	Prevents data leaks and helps your business system meet compliance requirements.

Security domain	Service name	Description
Security O&M service	On-premises security service	Helps you establish and optimize the cloud security system to protect your business system against attacks by using security features of Apsara Stack Security and other Apsara Stack services.

20.2. Benefits

As a pioneer of cloud security, Apsara Stack Security has received a variety of authoritative certifications. Through mature security systems and advanced security technologies, Apsara Stack Security can fully protect the security of the Apsara Stack environment.

Pioneer of cloud security

The Apsara Stack Security team has accumulated a wealth of security experience by protecting all internal business systems of Alibaba Group since 2005. Since its release in 2011, Apsara Stack Security has become a pioneer in providing comprehensive protection for cloud security.

Apsara Stack Security protects more than 40% of all websites in China. It prevents more than 50% of all distributed denial of service (DDoS) attacks and blocks up to 3.6 billion attacks every day. It has fixed over 6.13 million vulnerabilities over the last year.

Mature systems and advanced technologies

Apsara Stack Security is a service born from ten years of protection experience. After a decade of experience in providing security operations services for the internal business systems of Alibaba Group, Alibaba has obtained considerable security research achievements, security data, and security operations methods, and has built a professional cloud security team. Apsara Stack Security brings together the rich experience of these experts to develop the sophisticated systems that provide enhanced security for cloud computing platforms. This service can protect the cloud platform, cloud network environments, and cloud business systems of Apsara Stack users.

Comparison with traditional security products

Feature	Traditional security product	Apsara Stack Security
Comprehensive industry-leading security capabilities among Internet enterprises	A traditional security service provider only has limited products and features and cannot provide a comprehensive security protection system.	Alibaba has accumulated a large number of intelligence sources through years of attack prevention experience. This has allowed it to detect common Internet attacks including zero-day exploits, and provide comprehensive security capabilities.
Early risk detection	Traditional security service providers cannot detect risks due to a lack of complete monitoring systems.	Apsara Stack Security can detect and respond to critical vulnerabilities and security events quickly to prevent security issues.

Feature	Traditional security product	Apsara Stack Security
Security big data modeling analysis	Traditional security service providers cannot detect threats through signature scanning. The traditional log analysis feature only provides data collection and reporting. It does not provide data modeling analysis.	Big data modeling analysis enables Apsara Stack Security to detect threats in the entire network and display the security data. More than 30 algorithmic models are used to analyze the historical data, network data, and server data. This enables security situation awareness.
Scalability and decoupling with hardware	Traditional security products are developed based on the existing hardware devices. Security product software relies on the virtual machines created on virtualization platforms.	• Hardware and software decoupling: All modules are developed based on the cloud computing architecture and the common x86 hardware platform, and therefore do not rely on specific hardware. • Scalability: You can increase the amount of hardware for higher performance without the need to change the network architecture.
Collaboration between the network and servers	Traditional security service providers increase security features by adding devices. The devices can only collect device logs and status data and display the data on the management platform. They cannot collaborate to provide more features.	Apsara Stack Security provides complete Internet protection to guarantee the security of networks, applications, and servers. The security modules interact with each other to form a comprehensive protection system that blocks attacks effectively.
Compatibility with all data center environments and decoupling with specific cloud platforms	Most traditional security products are provided as hardware appliances. This makes the product incompatible with the cloud platforms based on Software Defined Network (SDN) technology.	Based on the interactions between servers and the operating system, Apsara Stack Security detects threats at the network perimeter through data analysis. This enables the service compatibility with all data center environments by avoiding the complex network topology inside the data centers.

20.3. Architecture

Apsara Stack Security consists of Apsara Stack Security Standard Edition and optional security services.

Apsara Stack Security Standard Edition

- Threat Detection Service (TDS)

This module collects network traffic and server information and detects possible vulnerability exploits, intrusions, and virus attacks through machine learning and data modeling. This module also provides you with up-to-date information about ongoing attacks to help you monitor the security status of your business.

- **Network Traffic Monitoring System**

This module is deployed on the network perimeter of Apsara Stack. This module allows you to inspect and analyze each inbound or outbound packet of an Apsara Stack network through traffic mirroring. The analysis results are used by other Apsara Stack Security modules.

- **Asset Vulnerability Monitoring**

This module analyzes known assets based on the built-in asset learning model to identify asset sources and help enterprises automatically detect unknown assets. This module also detects asset vulnerabilities to help enterprises identify unknown security risks in a timely manner.

- **Server Security**

This module collects information and performs detection by deploying clients on physical servers. This module monitors the security status of all physical servers in the Apsara Stack environment in real time, and provides a variety of features to help you detect security risks on physical servers in a timely manner. The features include Overview, Servers, Intrusion Detection, Server Fingerprints, and Log Retrieval.

- **Server Guard**

This module protects Elastic Compute Service (ECS) instances by providing security features such as vulnerability management, baseline check, intrusion detection, and asset management. To do this, the module performs operations such as log monitoring, file analysis, and signature scanning.

- **Web Application Firewall (WAF)**

This module protects web applications against common web attacks reported by Open Web Application Security Project (OWASP), such as Structured Query Language (SQL) injections, cross-site scripting (XSS), exploitation of vulnerabilities in web server plug-ins, trojan uploads, and unauthorized access. This module also blocks a large number of malicious requests to avoid data leaks and ensure both the security and availability of your websites.

Apsara Stack Security Standard Edition also provides on-premises security services. These services help you better use the features of Apsara Stack services such as Apsara Stack Security to ensure the security of your applications.

On-premises security services include pre-release security assessment, access control policy management, Apsara Stack Security configuration, periodic security check, routine security inspection, and urgent event handling. These services cover the entire lifecycle of your business in Apsara Stack and help you create a security operations system. This system enhances the security of your application systems and ensures both the security and stability of your business.

Optional security services

You can also choose the following service modules to enhance your system security.

- **Anti-DDoS Service**

This module detects and blocks distributed denial of service (DDoS) attacks.

- **Cloud Firewall**

This module sorts and isolates different types of business based on visualized business data to implement access control over east-west traffic in Apsara Stack.

- Sensitive Data Discovery and Protection (SDDP)

This module uses big data analytics capabilities and artificial intelligence (AI) technologies of Alibaba Cloud to detect and classify sensitive data based on your business requirements. This module can also mask sensitive data both in transit and at rest, monitor data flows, and detect abnormal activities. This module provides visible, controllable, and industry-compliant security protection for your sensitive data by using precise detection and analysis.

20.4. Features

20.4.1. Apsara Stack Security Standard Edition

Apsara Stack Security is developed based on the Apsara Stack environment and adopts a cloud security architecture that enables in-depth defense and multi-module collaboration. Unlike traditional software and hardware security products, Apsara Stack Security provides comprehensive and integrated cloud security protection at multiple layers, such as the network layer, application layer, and server layer.

The following table describes the features of Apsara Stack Security Standard Edition.

Features of Apsara Stack Security Standard Edition

Module	Feature	Description
Network Traffic Monitoring System	Traffic collection and analysis	Uses a bypass in traffic mirroring mode to collect inbound and outbound traffic that passes through the interconnection switch (ISW), and generates a traffic diagram.
	Malicious server identification	Detects attacks launched by internal servers and identifies the internal servers that launched the attacks.
	Abnormal traffic detection	Uses a bypass in traffic mirroring mode to detect abnormal traffic that has exceeded a specific threshold.
	Web application protection	Uses a bypass to block common attacks on web applications at the network layer based on default web attack detection rules.
Cloud Security Scanner	Asset discovery	Identifies asset sources based on known assets and the built-in asset learning model. This feature inspects assets on a regular basis to detect unknown assets and add them to the asset library.
	Asset management	Allows you to import, delete, group, export, query, and monitor assets and manage asset owners.
	Asset monitoring	Uses HTTP and ping commands to monitor assets, displays the details on the availability of an asset, and displays basic monitoring information of a monitored website based on custom alert rules.
	Vulnerability scanning	Scans your system for basic vulnerabilities, weak passwords, security vulnerabilities, and vulnerabilities reported by Common Vulnerabilities and Exposures (CVE), and supports automatic vulnerability inspection and baseline check.

Module	Feature	Description
	Vulnerability management	Automatically associates detected vulnerabilities with assets to visualize asset risks and help enterprises detect and manage risks at the earliest opportunity.
	External risk monitoring	Detects external risks based on features of employee behaviors and key enterprise information.
Server Guard	Baseline check	Performs security baseline checks for Elastic Compute Service (ECS) instances. The check items include accounts, weak passwords, and at-risk configuration items. The baseline checks ensure that the ECS instances comply with the security standards for enterprise servers.
	Vulnerability management	- Scans ECS instances for software vulnerabilities, and provides suggestions on vulnerability fixes. - Provides quick fixes for critical vulnerabilities in applications and operating systems on your ECS instances.
	Webshell detection and removal	Detects and removes webshells based on specified rules, and allows you to manually quarantine webshells.
	Brute-force attack blocking	Detects and blocks brute-force attacks in real time.
	Unusual logon alerting	Detects unusual logons based on the approved logon settings and generates alerts.
	Suspicious server detection	Detects suspicious activities such as reverse shells, Java processes running CMD commands, and unusual file downloads by using Bash.
	Asset fingerprints	Collects up-to-date information about the servers, such as ports, accounts, processes, and applications, to perform event tracking.
	Log retrieval	Centrally manages server logs of processes, networks, and system logons. This helps you to use logs to locate the cause of an issue.
	Detection overview	Provides the following modules: - Detection overview: provides statistics on protection for the last 24 hours and the last 30 days. - Access status monitor: displays the top 100 access requests in real time. - Export detection report: allows you to export daily reports, weekly reports, and scheduled task reports. - Attack detection statistics: provides statistics on attack detection.

Module	Feature	Description
Web Application Firewall (WAF)	Detection logs	Provides the following modules: • Attack detection logs: provides attack detection logs. The log list displays the processing results, attacked addresses, attack types, attacker IP addresses, and attack time. You can view log details about each attack. • HTTP flood detection logs: provides HTTP flood protection logs. The log list displays logs for matched HTTP flood protection rules, including the request URLs, the names of the matched rules, and the match time. You can filter logs based on the event generation time and the name of the HTTP flood protection rule. • System operations logs: provides system operations logs, including usernames, operations, and IP addresses. • Access logs: provides access logs, including the access addresses, destination IP addresses, source IP addresses, request methods, and response codes.

Module	Feature	Description
	Protection configuration	Provides the following modules: • Protection site management: allows you to create, delete, modify, enable, and disable feature forwarding proxies of a protected site. • Customized rules: allows you to create, delete, enable, and disable custom rules. This implements fine-grained HTTP access control for websites. • Website protection policies: ◦ Supports decoding methods, such as URL decoding, JSON parsing, Base64 decoding, hexadecimal conversion, backslash unescape, XML parsing, PHP deserialization, and UTF-7 decoding. ◦ Detects SQL injections, cross-site scripting (XSS), intelligence, cross-site request forgery (CSRF), server-side request forgery (SSRF), Hypertext Preprocessor (PHP) deserialization, Java deserialization, Active Server Pages (ASP) code injections, file inclusion attacks, file upload attacks, PHP code injections, command injections, crawlers, and server responses. ◦ Provides five built-in protection templates, including the template with default protection policies, monitoring mode template, anti-DDoS template, template for financial customers, and template for Internet customers. WAF allows you to customize the decoding algorithms in the templates, separately enable or disable each attack detection module, and configure the detection granularity. WAF also allows you to specify the Block Status Code parameter. ◦ Allows you to enable HTTP response detection and configure the length of the response body in detection rules. ◦ Allows you to configure the length of the request body in detection rules. ◦ Allows you to enable or disable detection timeout settings. • HTTP flood protection: allows you to configure access frequency control rules for domain names and URLs. This restricts the access frequency of IP addresses or sessions that meet the criteria, or blocks these IP addresses or sessions. WAF restricts the access frequency of known IP addresses or sessions or blocks these IP addresses or sessions. WAF supports the whitelist feature for HTTP flood protection. HTTP flood protection rules are not applicable to IP addresses or sessions in a whitelist.
	System management	Displays the workload, network, and detection statuses of a node. You can configure syslog to send logs and also configure the service- and system-related alert thresholds.

Module	Feature	Description
Threat Detection Service (TDS)	Overview	Provides a comprehensive security overview with statistics on security score, asset status, unhandled alerts, and handled alerts.
	Visualization	Displays the security data on the dashboard, including assets, vulnerabilities, baselines, attack sources, and attack distribution.
	Security alerts	Allows you to view and handle security events, including suspicious processes, webshells, unusual logons, sensitive file tampering, malicious processes, and suspicious network connections. Also provides threat detection for web applications.
	Attack analysis	Protects against common attacks on web applications and brute-force attacks.
	Cloud service check	Checks the security configurations of cloud services based on network access control and data security. This feature supports periodic checks that run automatically and manual checks. You can verify the check results or configure whitelist policies for the check results.
	Application whitelists	Allows you to add servers to the whitelist based on intelligent learning and identifies programs as trusted, suspicious, or malicious based on the whitelist. Unauthorized processes will be terminated.
	Assets	• Server: displays the security statuses for servers. You can view the numbers of all servers, servers at risk, unprotected servers, inactive servers, and new servers. • Cloud service: provides security status information for cloud services and supports Server Load Balancer (SLB) and NAT Gateway.
	Security reports	Allows you to query reports. For example, you can retrieve historical reports by report name.
Security Operations Center (SOC)	Dashboard	Allows security administrators to view the overall statistics and perform operations.
	Security monitoring	Allows you to view the security events of all users and the platform.
	Asset management	Allows you to view the security status of user assets and platform assets.
	Log analysis	Analyzes logs from multiple data sources, detects unexpected alerts, and improves alert detection of Apsara Stack.
	Report management	Allows security administrators to export reports.

Module	Feature	Description
	System configurations	Allows you to configure system features such as alerts, upgrades, global policies, and accounts.

20.4.2. On-premises security operations services

Apsara Stack Security Standard Edition provides on-premises security operations services that guarantee the security of your business systems.

The following table describes the on-premises security operations services that Apsara Stack Security provides.

On-premises security operations services

Category	Service	Description
Security operations	Asset research	Periodically researches your business systems in the cloud under your authorization and develops a business list containing information such as the business system name, Elastic Compute Service (ECS) information, Relational Database Service (RDS) information, IP address, domain name, and owner.
	New system assessment	• Detects system and application vulnerabilities in a new business system by using both automation tools and manual operations before you migrate the system to the cloud. • Provides suggestions and verification on vulnerability fixes.
	Periodic security assessment	• Periodically uses automation tools to detect system vulnerabilities, application vulnerabilities, and security risks in running business systems. • Provides suggestions on handling detected risks, including but not limited to security policy settings, patch updates, and application vulnerability handling.
	Access control management	Provides inspection and guidance on applying access control policies when a new business system is migrated to the cloud.
	Access control routine inspection	Periodically checks for access control risks of your business systems.
	Security risk routine inspection	Monitors and inspects security events in Apsara Stack Security, informs you of the verified events, and provides suggestions on event handling.
Apsara Stack Security maintenance	Rule update	Periodically updates the rules repository of Apsara Stack Security.
	Service integration	• Provides support for integrating Apsara Stack Security with your business systems. • Helps you customize and optimize security policies.

Category	Service	Description
Security event response	Event alerts	Synchronizes security events information from Alibaba Cloud, and helps you remove the risks.
	Event handling	Handles urgent events such as attacker intrusions.

20.4.3. Optional security services

The following table describes the optional security services that Apsara Stack Security provides.

Optional security services

Service	Feature	Description
Anti-DDoS Service	Traffic scrubbing against distributed denial of service (DDoS) attacks	Detects and defends against attacks, such as a SYN flood, ACK flood, ICMP flood, UDP flood, NTP flood, DNS flood, or HTTP flood.
	DDoS attack display	Allows you to search for DDoS attack events by IP address, status, and event information.
	DDoS traffic analysis	Allows you to monitor and analyze the traffic of a DDoS attack. You can also view the attack traffic protocol and the top 10 IP addresses that have launched the most attacks.
Cloud Firewall	Access control	- Centrally controls east-west traffic between servers and north-south traffic between the Internet and the servers. - Controls both inbound and outbound traffic. - Controls outbound traffic of external connections based on domain names. - Analyzes external connections to help you detect exceptions on Elastic Compute Service (ECS) instances.
	Real-time traffic monitoring	- Monitors outbound connections. - Analyzes traffic from the Internet to your ECS instances. - Presents asset information and access relationships in a visual display to help you detect abnormal traffic.

Service	Feature	Description
	Real-time defense	• Detects and blocks intrusions in real time. Analyzes the network traffic blocked by Cloud Firewall and IPS. • Synchronizes all malicious IP addresses detected across Alibaba Cloud and defends against potential threats, such as malicious visitors, scanners, and command-and-control servers. • Integrates the best practices of intrusion prevention policies implemented throughout Alibaba Cloud to ensure high accuracy of threat detection. • Supports installation-free virtual patches for business systems and protects against common vulnerabilities and zero-day and N-day vulnerabilities.
	Behavior backtracking	• Provides event logs to show threats or intrusions detected and blocked by IPS in real time. • Provides logs to record traffic that flows through Cloud Firewall. When a threat occurs, you can view traffic logs to analyze traffic, identify its source, and check configured access control policies. • Provides system operations logs to record all configurations and operations in Cloud Firewall.
Sensitive Data Discovery and Protection	Security situation overview	Allows you to view the overall security status of sensitive data.
	Detection and processing of suspicious activities	Detects suspicious activities related to sensitive data and allows you to confirm or exclude the activities after manual verification.
	Sensitive data detection	Detects sensitive data in services such as MaxCompute, Tablestore, Object Storage Service (OSS), AnalyticDB for MySQL, and ApsaraDB RDS.
	Static data masking	Uses data masking algorithms to mask sensitive data at rest.
	Intelligent audit	Creates audit rules to intelligently audit services such as OSS, MaxCompute, and ApsaraDB RDS.
	Data permission management	Displays departments and users by hierarchy, displays users and accounts by type, and allows you to query and manage detailed permissions of accounts.
	Dataflow monitoring	Allows you to view the dataflow details of DataHub and Data Integration.
	Rule configuration	Allows you to configure detection rules, risk levels, and abnormal output rules to detect sensitive data.

Service	Feature	Description
	Access authorization	Supports department-specific authorization and protects the data assets of authorized departments.

20.5. Restrictions

None

20.6. Terms

DDoS attacks

An attacker combines multiple computers by using the client-server model to form an attack platform and initiates a large number of valid requests to one or more targets from this platform to cause network failures. Distributed denial of service (DDoS) attacks are much stronger than common denial of service (DoS) attacks.

SQL injections

An attacker makes the server run malicious Structured Query Language (SQL) commands by inserting these commands in Web tables or inserting malicious strings in URL requests.

Traffic scrubbing

The traffic scrubbing service monitors the inbound traffic of a data center in real time and detects unusual traffic that may be from DDoS attacks and other attacks. This service scrubs the unusual traffic without affecting businesses.

Brute-force attacks

Brute-force attacks work by iterating through all possible combinations that can make up a password.

Webshells

A webshell is a script written in languages such as Active Server Pages (ASP) and Hypertext Preprocessor (PHP). Attackers can run a webshell on a Web server to perform risky operations. This enables attackers to obtain sensitive information or control the server through server penetration or privilege escalation.

Server intrusion detection

By analyzing server logs, Apsara Stack Security can detect attacks, such as system password cracking and logons from unusual IP addresses, and generate real-time alerts.

21.Key Management Service (KMS)

21.1. What is KMS?

Key Management Service (KMS) is a one-stop service platform for key management and data encryption. KMS provides simple, reliable, secure, and standard-compliant capabilities to encrypt and protect data. KMS greatly reduces your costs of purchase, operations and maintenance (O&M), and research and development (R&D) on cryptographic infrastructure and data encryption services. This helps you focus on the business development.

KMS provides the following features:

- Encryption key hosting

KMS supports encryption key hosting. An encryption key hosted on KMS is called a customer master key (CMK). You can manage the lifecycle of a CMK by enabling or disabling the CMK.

- BYOK

KMS supports Bring Your Own Key (BYOK). You can import your own keys to KMS to encrypt data on the cloud. This facilitates key management. You can import the following types of keys to KMS:

- Keys in your on-premises key management infrastructure (KMI)
- Keys in user-managed hardware security modules (HSMs) of Data Encryption Service

 **Note** Keys imported to managed HSMs in KMS cannot be exported by using any method because secure key exchange algorithms are used in KMS. Operators or third parties are not allowed to check the plaintext of keys.

- Automatic rotation of encryption keys

A CMK in KMS can have multiple key versions. Each version represents an independently generated key and does not have any relation with other versions. KMS automatically rotates encryption keys. This helps you implement the best security practices and comply with audit requirements. For more information, see the Overview and Automatic key rotation topics of *Key rotation* in *User Guide*.

- Fully managed HSMs

KMS provides fully managed HSMs. You can host keys in HSMs. Cryptographic operations are implemented in HSMs to protect key security.

 **Note** To use this feature, you must purchase an HSM and the KMS license of the Advanced edition.

- Simple cryptographic API operations

- KMS provides cryptographic API operations that are simpler than those for traditional cryptographic modules or cryptographic software libraries.
- Encryption keys in KMS support authenticated encryption with associated data (AEAD) and deliver additional authenticated data (AAD) to protect data integrity. For more information, see the EncryptionContext topic of *Use symmetric keys* in *User Guide*.

- CMK aliases

KMS allows you to create CMK aliases, which facilitate CMK usage. For more information, see the *Use aliases* topic in *User Guide*. For example, you can use CMK aliases to manually rotate CMKs in specific scenarios.

- Resource tags

KMS supports resource tags, which facilitate key resource management.

21.2. Benefits

KMS has advantages over traditional key management infrastructure (KMI), such as multi-service integration and ease of use.

Multi-service integration

KMS is integrated with multiple Alibaba Cloud services such as Elastic Compute Service (ECS), ApsaraDB RDS, and Object Storage Service (OSS). You can use CMKs in KMS to encrypt and control the data stored in these services and maintain control over the cloud computing and storage environments.

Ease of use

- Easy encryption

KMS simplifies abstract cryptographic concepts and provides cryptographic API operations that allow you to encrypt and decrypt data. For applications that require a key hierarchy, KMS provides convenient envelope encryption that implements a key hierarchy. For example, KMS generates a data key (DK) and uses a CMK as a key encryption key (KEK) to protect the DK. For more information, see the "Envelope encryption" topic in the "Features" chapter of *Technical White Paper*.

- Centralized key management

KMS provides centralized key management and control.

You can import keys from KMI or the HSMs of Data Encryption Service to KMS. No matter whether a key is imported from an external source or created in KMS, the confidential information or sensitive data in the key is used to encrypt data of an Apsara Stack service.

- BYOK

KMS supports the BYOK feature. You can import your own keys to KMS to encrypt data in the cloud. This facilitates key management. You can import the following types of keys to KMS:

- Keys in your on-premises KMI
- Keys in self-managed HSMs of Data Encryption Service

 **Note** Keys imported to managed HSMs in KMS cannot be exported by using any method because secure key exchange algorithms are used in KMS. Operators or third parties are not allowed to check the plaintext of keys.

- Custom key rotation policies

KMS supports automatic rotation of symmetric encryption keys based on security policies. You only need to configure a custom rotation cycle for a CMK. KMS automatically generates new CMK versions. A CMK can have multiple key versions. Each version can be used to decrypt corresponding ciphertext data. The latest key version, which is called the primary version, is an active encryption key and is used to encrypt current data. For more information, see the "Automatic key rotation" topic in the "Key rotation" chapter of *User Guide*.

21.3. Scenarios

This topic describes the common scenarios in which you can use KMS.

Typical scenarios

Role	Demand	Scenario
Application developer	Protect the security of sensitive data in applications.	Sensitive data is used in applications. Application developers need to use encryption keys to encrypt sensitive data and use KMS to protect the encryption keys.

Encryption methods

- Envelope encryption

Your CMKs are stored in KMS. You only need to deploy enveloped data keys (EDKs). You can call the Decrypt API operation to decrypt the EDKs and use the returned plaintext DKs to encrypt or decrypt your local business data.

For more information about envelope encryption, see the Envelope encryption topic of *Features in Technical White Paper*.

- Direct encryption

You can call the Encrypt or Decrypt API operation to directly encrypt or decrypt sensitive data by using CMKs.

- Server-side encryption

If you use Alibaba Cloud services to store data, you can use the server-side encryption feature of these services to encrypt and protect data in a simple and effective way. For example, you can use the server-side encryption feature of OSS to protect buckets that store sensitive data or use transparent data encryption (TDE) to protect tables that store sensitive data.

21.4. Terms

This topic introduces the terms used in KMS.

 **Note** For more information about API operations mentioned in this topic, see *API reference* in *Developer Guide*.

Key Management Service (KMS)

KMS provides features such as key hosting and cryptographic operations. KMS implements security practices such as key rotation and can be integrated with other Alibaba Cloud services to encrypt the user data managed by these services. KMS frees you from the need to manually maintain the security, integrity, and availability of your keys. You only need to focus on data encryption, data decryption, and digital signature generation and verification based on your business requirements.

customer master key (CMK)

A CMK is used to encrypt data keys (DKs) and generate enveloped data keys (EDKs). It can also be used to encrypt a small volume of data. You can call the CreateKey API operation of KMS to create a CMK.

envelope encryption

To encrypt business data, you can call the GenerateDataKey or GenerateDataKeyWithoutPlaintext operation to generate a symmetric key and encrypt the symmetric key by using a CMK. An EDK is returned. The EDK is secure even if it is stored and transferred over unsecured communication channels. If you want to use the symmetric key, you only need to call the Decrypt operation to decrypt the EDK.

For more information about envelope encryption, see the Envelope encryption topic of *Features in Technical White Paper*.

data key (DK)

A DK refers to a plaintext key used to encrypt data.

 **Note** You can call the GenerateDataKey operation to generate a DK and encrypt the DK by using a CMK. The plaintext and ciphertext of the DK are returned. In a general sense, DK refers to the plaintext and EDK refers to the ciphertext.

enveloped data key (EDK)

An EDK refers to a ciphertext DK generated by using envelope encryption.

 **Note** If the plaintext of a DK is not needed, you can call the GenerateDataKeyWithoutPlaintext operation to return only the ciphertext of the DK.

hardware security module (HSM)

An HSM is a hardware device that performs cryptographic operations and securely generates and stores keys. The Managed HSM feature provided by KMS meets both the testing and validation requirements of regulatory agencies and provides you with high security for your keys managed in KMS.

EncryptionContext

EncryptionContext refers to the encapsulation of **authenticated encryption with associated data (AEAD)** in KMS. KMS uses imported encryption context as the additional authenticated data (AAD) of the symmetric encryption algorithm for cryptographic operations. In this way, KMS provides additional integrity and authenticity for encrypted data. For more information, see the EncryptionContext topic of *Use symmetric keys* in *User Guide*.

22.Apsara Stack DNS

22.1. What is Apsara Stack DNS?

Apsara Stack DNS is a service that runs on Apsara Stack to resolve domain names over internal networks, such as VPCs, data centers, and the classic network. You can configure rules to map domain names to IP addresses. Apsara Stack DNS then distributes domain name requests from clients to cloud resources, user-created business applications, business systems on your internal networks, or the business resources of Internet service providers (ISPs).

Apsara Stack DNS provides the DNS resolution and Global Server Load Balancer (GSLB) services in VPCs, data centers, and the classic network. You can perform the following operations by using Apsara Stack DNS in these internal networks:

- Access other ECS instances deployed in the same VPC.
- Access other cloud service instances on Apsara Stack.
- Access enterprise business systems.
- Access services over the Internet.
- Use the GSLB service to implement multiple-active solutions and disaster recovery, such as local active-active, local multi-active, remote active-active, active geo-redundancy, and geo-disaster recovery.
- Connect to Apsara Stack DNS with your own DNS servers over a leased line.

22.2. Edition comparison

This topic describes the differences between Apsara Stack DNS editions.

 **Note** In Apsara Stack Enterprise, four DNS editions are supported: DNS Lightweight Basic Edition, DNS Basic Edition, DNS Standard Edition, and Internal GTM Standard Edition. In Apsara Stack Agility, two DNS editions are supported: DNS Basic Edition and Internal GTM Standard Edition.

Feature category	Feature	DNS Lightweight Basic Edition	DNS Basic Edition	DNS Standard Edition	Internal GTM Standard Edition
Internal DNS resolution management	Global basic DNS resolution	Supported	Supported	Supported	Supported
	Global load balancing (weight)	Supported	Supported	Supported	Supported
	Global domain name forwarding	Supported	Supported	Supported	-
	Global default forwarding	Supported	Supported	Supported	-

Feature category	Feature	DNS Lightweight Basic Edition	DNS Basic Edition	DNS Standard Edition	Internal GTM Standard Edition
	Internet recursive resolution	Not supported by default. (However, it can be supported if you use the NAT Gateway solution.)	Supported	Supported	-
PrivateZone (tenant isolation)	Tenant-based basic DNS resolution	Not supported	Not supported	Supported	-
	Tenant load balancing (weight)	Not supported	Not supported	Supported	-
	Tenant domain name forwarding	Not supported	Not supported	Supported	-
	Tenant-based default forwarding	Not supported	Not supported	Supported	-
	VPC binding and unbinding	Not supported	Not supported	Supported	-
Internal Global Traffic Manager	Scheduling instance management	-	-	-	Supported
	Address pool management	-	-	-	Supported
	Scheduling domain management	-	-	-	Supported
	Data synchronization management of clusters (multi-cloud)	-	-	-	Supported
	Global data synchronization (multi-cloud)	-	-	-	Supported

Feature category	Feature	DNS Lightweight Basic Edition	DNS Basic Edition	DNS Standard Edition	Internal GTM Standard Edition
Others	Independent physical machine deployment	Not required	Required	Required	Required. It must be deployed on the physical machines of the DNS Basic Edition or DNS Standard Edition.
	Web UI console	Supported	Supported	Supported	Supported
	Upgrade to DNS Basic Edition	Supported (New physical machines must be deployed.)	-	-	-
	Upgrade to DNS Standard Edition	Supported (New physical machines must be deployed.)	Supported	-	-

22.3. Benefits

Enterprise domain name management

Apsara Stack DNS provides management and resolution services for your domain names. It supports the following features:

- Performs forward and reverse DNS resolutions for domain names of cloud service instances, such as ECS instances.
- Performs forward and reverse DNS resolutions for your internal domain names.
- Allows you to add, modify, and delete DNS records of the following types: A, AAAA, CNAME, NS, MX, TXT, SRV, and PTR.
- Allows you to add multiple A, AAAA, or PTR records at a time. DNS servers randomly respond to all DNS queries through round robin to achieve load balancing.

Flexible integration with data centers

Apsara Stack DNS can forward enterprise domain names and provide the following services for you to flexibly build your network and cascade DNS servers with user-created DNS servers:

- Global default forwarding
- Forwarding queries for specific domain names

Internet access from enterprise servers

Apsara Stack DNS supports recursive resolution for Internet domain names, which allows your servers to access the Internet.

Tenant isolation (DNS Standard Edition only)

Apsara Stack DNS allows you to manage private zones in VPCs, resolve internal domain names, and isolate DNS records by tenant.

- You can add, delete, modify, and query private authoritative zones. You can also bind and unbind private authoritative zones to and from VPCs.
- You can add, delete, modify, and query private forwarding zones. You can also bind and unbind private forwarding zones to and from VPCs.

GSLB

Global Server Load Balancer (GSLB) provides the following features on internal networks:

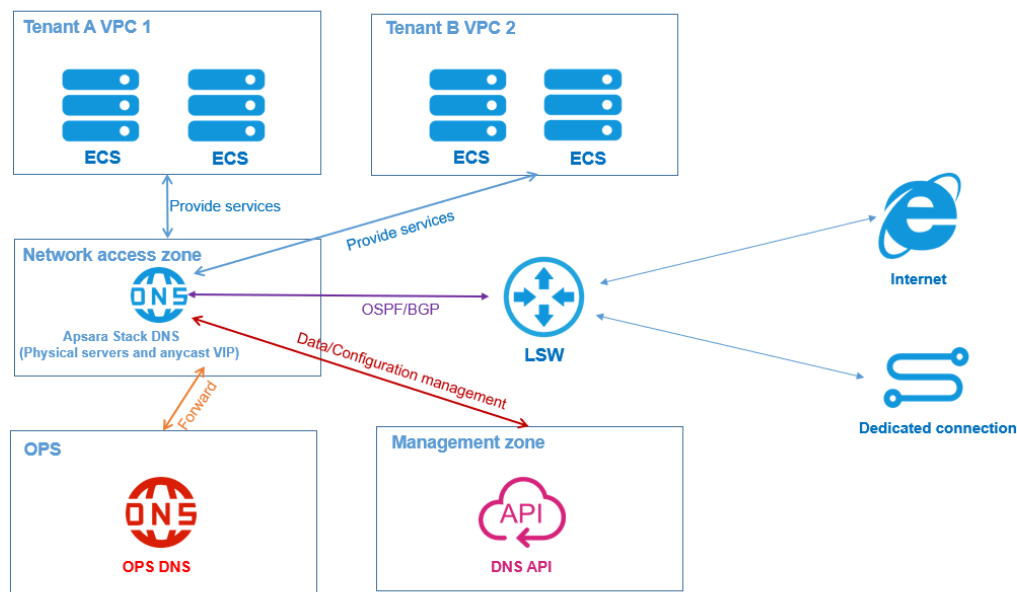
- Allows you to add multiple A, AAAA, or CNAME records at a time. DNS servers respond to DNS queries based on the weight of each record type to achieve load balancing.
- Supports scheduling line management. You can customize lines and their priorities to allow clients to access the nearest servers and implement intelligent traffic scheduling based on geographical locations and application groups. This accelerates access to applications.
- Synchronizes configuration data for resolution among multiple clusters for which GSLB is activated. This feature is supported in multi-cloud scenarios.
- Supports address pool management to centrally manage enterprise applications by application service cluster.
- Supports custom global scheduling domains. You can centrally manage and code global scheduling instances based on your naming conventions.

Centralized management console

You can access DNS and other cloud services in the Apsara Uni-manager Management Console with one account. This implements web-based data and service management, which simplifies operations.

22.4. Architecture

Architecture of Apsara Stack DNS



Note Different from Apsara Stack Enterprise, Apsara Stack Agility allows you to use ZStack to create computing resources and VPCs. The architecture of Apsara Stack DNS in Apsara Stack Enterprise is different from that in Apsara Stack Agility. The following sections describe the architecture in the two editions.

Apsara Stack Enterprise

Architecture of Apsara Stack DNS (DNS Basic Edition and DNS Standard Edition)

- Uses two independent physical machines that are deployed in the network access zone to improve service availability. Apsara Stack DNS in this architecture can be scaled in or out.
- Issues anycast virtual IP address (VIP) routing requests over the LAN switch (LSW) by using Open Shortest Path First (OSPF) or Border Gateway Protocol (BGP). Anycast VIPs provide DNS services for tenants in VPCs or in the classic network. The outbound IP address configured on the DNS servers can be used to forward requests to the OPS DNS server, Internet, or a dedicated enterprise network based on forwarding and recursive rules.
- Manages data and configurations by using APIs in the management zone.
- Allows you to create and query domain names on a web UI, forwards requests for cloud service domain names to the OPS DNS server, performs recursive DNS queries for Internet domain names, allows you to add, modify, delete, and query authoritative domain names and forwarding domain names of private zones, and binds and unbinds a private zone to and from a VPC.

Architecture of Apsara Stack DNS (DNS Lightweight Basic Edition)

- Supports the deployment with two physical machines on the OPS3 or OPS4 base, which eliminates the need to apply for an independent physical machine. The two physical machines achieve high availability. Apsara Stack DNS in this architecture cannot be scaled in or out.
- Issues anycast VIP routing requests over the LSW by using OSPF or BGP. Anycast VIPs provide DNS services for VPCs and the classic network of tenants. The outbound IP address configured on the DNS servers can be used to forward requests to the OPS DNS server, Internet, or a dedicated enterprise network based on forwarding and recursive rules.

- Manages data and configurations by using APIs in the management zone.
- Allows you to create and query domain names on a web UI, forwards requests for cloud service domain names to the OPS DNS server, and performs recursive DNS queries for Internet domain names.

Architecture of Apsara Stack DNS (internal GTM Standard Edition)

- Depends on the deployment of DNS Basic Edition or DNS Standard Edition. Apsara Stack DNS of the internal GTM Standard Edition is deployed on the two physical machines of DNS Basic Edition or DNS Standard Edition in the network access zone. Apsara Stack DNS in this architecture can be scaled in or out.
- Issues anycast VIP routing requests over the LSW by using OSPF or BGP. Anycast VIPs provide DNS services for VPCs and the classic network of tenants.
- Manages data and configurations by using APIs in the management zone.
- Allows you to manage domain names on a web UI, allows you to add, modify, delete, and query address pools, access policies, and scheduling instances. You can also create and delete Global Traffic Manager (GTM) synchronization clusters.

Apsara Stack Agility

Architecture of Apsara Stack DNS (DNS Basic Edition)

- Uses two independent physical machines that are deployed in the network access zone to improve service availability. Apsara Stack DNS in this architecture can be scaled in or out.
- Issues anycast VIP routing requests over the LSW by using OSPF or BGP. Anycast VIPs provide DNS services for VPCs and the classic network of tenants. The outbound IP address configured on the DNS servers can be used to forward requests to the OPS DNS server, Internet, or a dedicated enterprise network based on forwarding and recursive rules.
- Manages data and configurations by using APIs in the management zone.
- Allows you to create and query domain names on a web UI, forwards requests for cloud service domain names to the OPS DNS server, and performs recursive DNS queries for Internet domain names.

22.5. Features

1. Internal DNS resolution management

Internal DNS resolution management allows you to manage global internal domain names, global forwarding configurations, and global recursive resolution configurations that you have created in Apsara Stack. Changes to these configurations take effect on all VPCs and the classic network.

This feature provides the same global DNS resolution service to all servers in VPCs. DNS servers use anycast IP addresses within a region. This way, seamless service failover and failback can be achieved in a specific region where data centers support disaster recovery. Note: If you do not need to upgrade Apsara Stack DNS to the Standard Edition, you can configure DNS server addresses as global anycast IP addresses to implement seamless service failover and failback over the entire network if data centers support disaster recovery.

1.1 Global internal domain names

Allows you to register, search, and delete global internal domain names and add descriptions for these domain names. You can also add, delete, and modify DNS records. The following DNS record types are supported: A, AAAA, CNAME, MX, PTR, TXT, SRV, NAPTR, CAA, and NS.

Allows you to add multiple A, AAAA, or PTR records at a time. DNS servers randomly respond to all DNS queries through round robin to achieve load balancing.

Allows you to add multiple A, AAAA, or CNAME records at a time. DNS servers respond to DNS queries based on the weight of each record type to achieve load balancing.

1.2 Global forwarding configurations

Forwards domain name requests to another DNS server for resolution.

Supports global default forwarding, which forwards requests of domain names that do not have forwarding configurations to another DNS server for resolution.

Apsara Stack DNS can forward requests with or without recursion.

- **Forward All Requests (without Recursion):** Only the specified DNS server is used to resolve domain names. If the resolution fails or the request times out, a message is returned to the DNS client to indicate that the query failed.
- **Forward All Requests (with Recursion):** The specified DNS server is preferentially used to resolve domain names. If the resolution fails, the local DNS server is used.

1.3 Global recursive configurations

Supports recursive resolution for Internet domain names, which enables your servers to access the Internet.

Allows you to enable, disable, or modify the global default forwarding configurations.

2. PrivateZone (DNS Standard Edition only)

The PrivateZone feature allows you to create tenant-specific domain names in VPCs. You can bind and unbind the domain names to and from VPCs as required to isolate tenants. Changes to these configurations take effect only in the VPCs to which the domain names are bound.

This feature provides personalized DNS resolution service to servers in the VPCs to which the domain names are bound. DNS servers use anycast IP addresses within a region. This way, seamless service failover and fallback can be achieved in a specific region where data centers support disaster recovery.

2.1 Tenant internal domain names

Allows you to register, search, and delete tenant internal domain names and add descriptions for these domain names. You can also add, delete, and modify DNS records. The following DNS record types are supported: A, AAAA, CNAME, MX, PTR, TXT, SRV, NAPTR, CAA, and NS.

Allows you to add multiple A, AAAA, or PTR records at a time. DNS servers randomly respond to all DNS queries through round robin to achieve load balancing.

Allows you to add multiple A, AAAA, or CNAME records at a time. DNS servers respond to DNS queries based on the weight of each record type to achieve load balancing.

Allows you to bind and unbind a domain name to and from a VPC.

2.2 Tenant forwarding configurations

Forwards domain name requests to another DNS server for resolution.

Supports global default forwarding, which forwards requests of domain names that do not have forwarding configurations to another DNS server for resolution.

Apsara Stack DNS can forward requests with or without recursion.

- **Forward All Requests (without Recursion):** Only the specified DNS server is used to resolve domain names. If the resolution fails or the request times out, a message is returned to the DNS client to indicate that the query failed.

- **Forward All Requests (with Recursion):** The specified DNS server is preferentially used to resolve domain names. If the resolution fails, the local DNS server is used.

Allows you to bind and unbind a domain name to and from a VPC.

3. Internal Global Traffic Manager (internal GTM Standard Edition only)

Internal Global Traffic Manager (GTM) provides multi-cloud disaster recovery for your domain names. You can connect your domain names to an internal GTM instance to manage traffic loads between Apsara Stack systems.

Internal GTM supports internal Global Server Load Balancer (GSLB). This feature intelligently allocates IP addresses for DNS queries from request sources based on configured scheduling policies. It also supports multi-cloud, hybrid deployment and configuration data synchronization between cloud networks.

3.1 Scheduling instance management

Allows you to manage scheduling instances. Each scheduling instance corresponds to an application instance.

Allows you to manage address pools. Each address pool corresponds to a service cluster of an application instance.

Allows you to manage scheduling domains and set the scheduling domains to which scheduling instances belong. You can centrally manage and code global scheduling instances based on your own naming conventions.

3.2 Scheduling line management

Supports scheduling line management. You can customize lines and their priorities to allow clients to access the nearest nodes and implement intelligent traffic scheduling based on geographical locations and application groups. This accelerates access to applications.

3.3 Data synchronization management

Allows you to manage global data synchronization links. You can create data synchronization links, manage data synchronization configurations, and view data synchronization information of multiple internal GTM services. The information includes local system information, information of cluster nodes on which data synchronization relationship has been established, and primary and secondary relationships.

Allows you to manage the messages for changes to data synchronization links, which helps you confirm request messages for primary nodes to actively add secondary nodes.

22.6. Scenarios

Apsara Stack DNS is a key network service that controls data traffic for Apsara Stack. It resolves domain names, balances server loads, and connects Apsara Stack with data centers and Alibaba Cloud public cloud. Apsara Stack DNS provides a complete suite of solutions to deploy a cloud environment, achieve high availability and disaster recovery for data centers, and balance server loads to secure your IT services.

Apsara Stack DNS provides four categories of solutions in the following eleven scenarios for enterprise users:

 **Note** Different from Apsara Stack Enterprise, Apsara Stack Agility allows you to use ZStack to create computing resources and VPCs.

(1) Basic DNS resolution (DNS Basic Edition, DNS Agility Edition, or DNS Lightweight Basic Edition)

Scenario 1: Access cloud resource instances over a VPC

You can use Apsara Stack DNS to access ApsaraDB RDS, Server Load Balancer (SLB), and Object Storage Service (OSS) instances from ECS or Docker instances over a VPC.

Scenario 2: Access ECS or Docker instances by using hostnames over a VPC

You can use Apsara Stack DNS to create hostnames for your ECS or Docker instances, and access and manage them by using hostnames over a VPC.

Scenario 3: Access internal service domain names over a VPC

You can use Apsara Stack DNS to develop your own PaaS or SaaS services on Apsara Stack and access the PaaS or SaaS services by using domain names over a VPC.

Scenario 4: Schedule internal service requests on Apsara Stack by using the round robin algorithm

If the SaaS service in scenario 3 is deployed in multiple data centers or regions, you can use Apsara Stack DNS to access the service and evenly distribute requests to different nodes in a VPC.

Scenario 5: Access Internet services over a VPC or the classic network

You can use Apsara Stack DNS to access Internet services from a VPC or the classic network.

Scenario 6: Establish connections between Apsara Stack and other networks

You can use Apsara Stack DNS to establish connections between Apsara Stack and other networks, such as internal networks, Alibaba Cloud public cloud, and other external networks by using domain names.

(2) Tenant isolation (DNS Standard Edition)

Scenario 7: Isolate tenant resources on Apsara Stack

You can use Apsara Stack DNS to isolate tenant resources on Apsara Stack so that the internal DNS resolution data and the default forwarding configurations of each tenant are invisible to other tenants. You can configure your private DNS resolution data to complete business addressing and scheduling.

Scenario 8: Establish connections among global resources on Apsara Stack

If you need to allow all tenants to share global resources and configurations on Apsara Stack, system administrators can configure global DNS resolution data and configurations to complete business addressing and scheduling.

Scenario 9: Perform VPC-based intelligent scheduling on Apsara Stack

You can use Apsara Stack DNS to provide different DNS records for different VPCs of a tenant based on the same host record.

(3) Global scheduling (internal GTM Standard Edition)

Scenario 10: Schedule the traffic loads of internal network services on Apsara Stack based on weights

If the PaaS or SaaS service in scenario 3 is deployed in multiple data centers or regions, you can use Apsara Stack DNS to access the service from within or outside the cloud, and distribute requests to different nodes in backend service clusters based on the weights of these nodes.

Scenario 11: Schedule the traffic loads of internal network services on Apsara Stack based on the geographical locations or application groups of the request sources

If the PaaS or SaaS service in scenario 3 is deployed in multiple data centers or regions, you can use Apsara Stack DNS to access the service from within or outside the cloud, and distribute requests to different nodes based on the geographical locations or application groups of the request sources.

(4) Enterprise disaster recovery (internal GTM Standard Edition)

Scenario 11: Schedule the traffic loads of internal network services on Apsara Stack to achieve disaster recovery

If the PaaS or SaaS service in scenario 3 is deployed in multiple data centers or regions, you can use Apsara Stack DNS to access the service from within or outside the cloud, and distribute internal access requests from backend cluster A (primary site) to backend cluster B (secondary site) in disaster recovery scenarios.

22.7. Limits

This topic describes limits on Apsara Stack DNS clusters.

DNS Basic Edition

Cluster	Module	Server type	Minimum configuration	Quantity

Cluster	Module	Server type	Minimum configuration	Quantity
Access cluster	Resolution	Q46S1.2C	16-core CPU, 96 GB of memory, 600 GB of hard disk space, two GE ports, and four 10GE ports. The downgraded Q46 model is used, and the network must support the Q46S1.2C server type.	2
Management cluster	Management	Container	4-core CPU, 8 GB of memory, 60 GB of hard disk space, and network connections.	2

DNS Standard Edition

Cluster	Module	Server type	Minimum configuration	Quantity
Access cluster	Resolution	Q46S1.2C	40-core CPU (two Intel Xeon Silver 4114 CPUs), 192 GB of memory, 1,200 GB of hard disk space, two GE ports, and four 10GE ports.	2
Management cluster	Management	Container	4-core CPU, 8 GB of memory, 60 GB of hard disk space, and network connections.	2

DNS Lightweight Basic Edition

Cluster	Module	Server type	Minimum configuration	Quantity
Access cluster	Resolution	OPS3 or OPS4	8-core CPU, 16 GB of memory, 480 GB of hard disk space, and two GE ports.	2

Cluster	Module	Server type	Minimum configuration	Quantity
Management cluster	Management	Container	4-core CPU, 8 GB of memory, 60 GB of hard disk space, and network connections.	2

 **Note** Tenant-related features are not supported in this edition. You must plan two anycast IP addresses for DNS resolution of ECS instances. By default, the resolution module of this edition provides DNS resolution for ECS instances in new deployment scenarios since Apsara Stack DNS V3.11. You must plan a source IP address for forwarding DNS queries.

22.8. Terms

DNS

Domain Name System (DNS) is a distributed database that is used for TCP/IP applications. It translates domain names into IP addresses, and selects paths for emails.

Domain name resolution

Domain name resolution maps domain names to IP addresses by using the DNS system. It includes both authoritative DNS and recursive DNS.

Recursive DNS

Recursive DNS queries domain names cached on the local DNS server or sends a request to the authoritative DNS system to obtain the corresponding IP addresses. You can use recursive DNS to resolve Internet domain names.

Authoritative DNS

Authoritative DNS resolves the names of root domains, top-level domains, and various other domains.

Authoritative domain names

Authoritative domain names are domain names resolved by the local DNS server. You can configure and manage the domain name resolution data on the local DNS server.

DNS forwarding

DNS forwarding uses two DNS servers to provide DNS resolution services. The local DNS server is used to configure and manage the domain name resolution data. The other DNS server is used to resolve domain names.

Default forwarding

If DNS queries for authoritative domain names are not resolved by the local DNS server, they are forwarded to another DNS server for resolution.

23. API Gateway

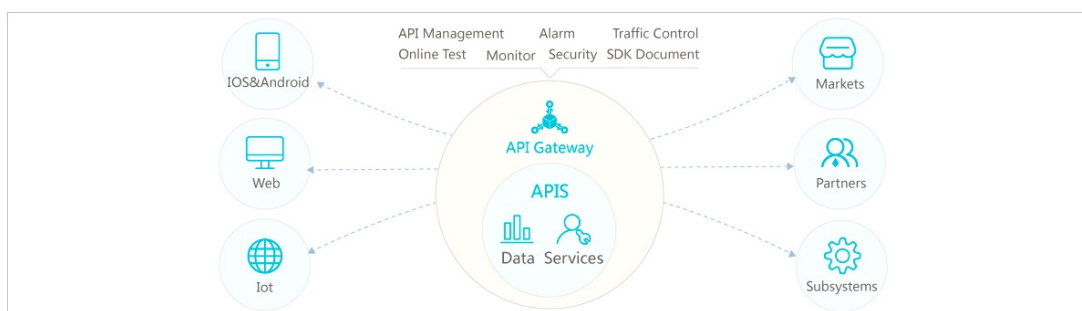
23.1. What is API Gateway?

API Gateway provides a comprehensive suite of API hosting services that help you share capabilities, services, and data with partners in the form of APIs.

- API Gateway provides multiple security mechanisms to secure APIs and reduce the risks arising from open APIs. These mechanisms include protection against replay attacks, request encryption, identity authentication, permission management, and throttling.
- API Gateway provides API lifecycle management that allows you to define, publish, and unpublish APIs. This improves API management and iteration efficiency.

API Gateway allows enterprises to reuse and share their capabilities with each other so that they can focus on their core business.

API Gateway



23.2. Features

API lifecycle management

- Manages APIs throughout their entire lifecycle, including publishing, testing, and unpublishing APIs.
- Supports maintenance features such as routine management, version management, and quick rollback.

Comprehensive security protection

- Supports multiple authentication methods and HMAC (SHA-1 and SHA-256) algorithms.
- Supports HTTPS and SSL encryption.
- Provides multiple security mechanisms to prevent injections, replay attacks, and tampering.

Flexible access control

- API Gateway implements access control on apps that are used to make API requests.
- Only authorized apps can call APIs.
- API providers can authorize apps to call APIs.

Various plug-in features

- Allows you to use various plug-ins to expand the features of APIs in a pluggable manner.
- Provides various plug-ins, including throttling, IP address-based access control, backend signature,

JWT authentication, cross-origin resource sharing (CORS), caching, backend routing, access control, circuit breaker, and error code mapping.

Request verification

Supports the verification of parameter types and values (range, enumerated value, and regular expression). Requests with invalid parameter types or values are denied by API Gateway. This reduces backend resources wasted on invalid requests and significantly reduces backend service processing costs.

Data conversion

Enables you to configure mapping rules to translate frontend and backend data.

- Supports data conversion for frontend requests.

Automated tools

- Automatically generates API documentation.
- Provides SDK samples in multiple programming languages.
- Provides graphical debugging tools for quick testing and deployment.

Monitoring and alerting

- Provides a graphical real-time API monitoring panel that displays information such as the number of API calls, response time, and error rate.
- Allows you to configure alerts and to track the operating status of each API in real time.
- Supports API-based full log query by using Log Service.

23.3. Benefits

Easy maintenance

After you create APIs in API Gateway, API Gateway performs all the other API management functions. This significantly reduces routine maintenance costs.

High request processing performance

API Gateway uses a distributed deployment and auto-scaling model to respond to a large number of API call requests at very low latency. It provides secure and efficient gateway functions for your backend services.

Security and stability

Your services are open to API Gateway only over an internal network. API Gateway also provides enhanced permission management functions and precise request throttling functions. It makes your services secure, stable, and controllable.

Comprehensive functionality

API Gateway supports API lifecycle management and provides graphical debugging tools and various plug-ins. It can also automatically generate SDKs of different programming languages. This meets your requirements on API usage.

23.4. Terms

Before you use API Gateway, familiarize yourself with the following basic concepts.

app

An app defines the identity of an API caller. To call an API, you must create an app first.

AppKey and AppSecret

Each app has an AppKey and AppSecret pair. This pair is encrypted and attached to a request as the signature.

encrypted signature

An encrypted signature is attached to each API request and is authenticated by API Gateway.

authorization

The API service provider grants an app the permission to call an API. Only authorized apps can call specified APIs.

API lifecycle

The API service provider manages an API in stages, including the creation, test, publish, unpublish, and version change stages.

API definition

An API definition is a set of rules defined by the API service provider during the creation of an API. The API definition specifies the backend service, request format, receive format, and return format.

parameter mapping

Parameter mapping is configured by the API service provider. It is used when the parameters in a request are inconsistent with those of the API backend service.

parameter verification

Parameter verification is performed based on a set of rules defined by the API service provider. API Gateway filters out invalid requests based on these rules.

constant parameter

API users do not need to enter the constant parameters. The constant parameters are always received by the backend service.

system parameter

API Gateway includes system parameters such as CaClient IP (request IP address) in the requests sent to your backend service.

API group

An API group is a group of APIs that are managed by the API service provider as a whole. Before you create an API, you must create an API group.

second-level domain

A second-level domain is a domain name that you bind to an API group when you create the group. This domain name is used to test API calling.

independent domain

An independent domain is a domain name that you bind to an API group when you open an API in the group. You must access the independent domain to call an API.

plugin

A plugin is a pluggable extension used to implement API functions by performing binding and unbinding operations.

signature key

A signature key is created by the API service provider and bound to an API. The signature information is added to each request sent from API Gateway to the backend service. The backend service checks the signature information to ensure security.

throttling policy

The API service provider can configure a throttling policy to limit the maximum number of requests for an API, and the maximum number of API requests that can be initiated by a user or an app. The throttling granularity can be day, hour, or minute.

24.MaxCompute

24.1. What is MaxCompute?

MaxCompute is a highly efficient, highly available, and low-cost EB-level computing service for big data. It is independently developed by Alibaba Cloud. This service is used within Alibaba Group to process exabytes of data each day. MaxCompute is a distributed system designed for big data processing. As one of the core services in the Alibaba Cloud computing solution, MaxCompute is used to store and compute structured data.

MaxCompute is designed to support multiple tenants and provide features such as data security and horizontal scaling. It provides a centralized graphical user interface (GUI) and centralized APIs for various data processing tasks of different users based on an abstract job processing framework.

MaxCompute is used to store and compute large amounts of structured data at a time. It provides various data warehousing solutions as well as big data analytics and modeling services. MaxCompute aims to provide easy analysis and processing of large amounts of data. You can analyze big data without a deep knowledge of distributed computing.

MaxCompute has the following features:

- Uses a distributed architecture that can be horizontally scaled based on your business requirements.
- Provides automatic storage and fault tolerance mechanisms to ensure high data reliability.
- Allows all computing tasks to run in sandboxes to ensure high data security.
- Uses RESTful APIs to provide services.
- Supports both uploads and downloads of high-concurrency, high-throughput data.
- Supports two service models: the offline computing model and the machine learning model.
- Supports data processing methods based on programming models such as SQL, MapReduce, Graph, and MPI.
- Supports multiple tenants, which allows multiple users to collaborate on data analysis.
- Manages user permissions based on access control lists (ACLs) and policies, which allows you to flexibly configure access control policies to prevent unauthorized data access.
- Supports Spark on MaxCompute, the enhanced Spark application.
- Supports Elasticsearch on MaxCompute, the enhanced Elasticsearch application.
- Supports the access to and processing of unstructured data.
- Supports the deployment of multiple clusters in a single region.
- Supports multi-region deployment.
- Uses the column store method and supports Key Management Service (KMS) to encrypt data files. Allows you to encrypt data based on your business requirements. Allows you to encrypt all data or only critical data.
- Stores audit logs and automatically dumps them to a specific server directory for long-term storage and management.

24.2. Integration with other Alibaba Cloud services

MaxCompute has been integrated with some other Alibaba Cloud services to quickly implement a variety of business scenarios.

MaxCompute and DataWorks

DataWorks uses MaxCompute as the core computing and storage engine to provide offline processing and analysis of large amounts of data. DataWorks offers fully hosted services for visual workflow development, scheduling, and operations and maintenance (O&M).

MaxCompute works with DataWorks to provide complete extract, transform and load (ETL) and data warehouse management capabilities, as well as classic distributed computing models such as SQL, MapReduce, and Graph. These models enable you to process large amounts of data, which reduces costs and ensures data security.

MaxCompute and Data Integration

You can use Data Integration to load data from different sources such as MySQL databases into MaxCompute, and export data from MaxCompute to various business databases.

Data Integration has been integrated into DataWorks and is configured and runs as a data synchronization task. You can directly add MaxCompute data sources to DataWorks and configure tasks to read or write data from or to MaxCompute tables. The entire process is completed on a single platform.

MaxCompute and PAI

Machine Learning Platform for AI (PAI) is a machine learning algorithm platform based on MaxCompute. PAI provides an end-to-end machine learning platform for data processing, model training, service deployment, and prediction without data migration. After you create a MaxCompute project and activate PAI, you can use the algorithm components of PAI to perform operations such as model training on MaxCompute data.

MaxCompute and Quick BI

After you process data in MaxCompute, you can add projects as Quick BI data sources. Then, you can create reports based on MaxCompute table data in the Quick BI console for visual data analysis.

MaxCompute and AnalyticDB for MySQL

AnalyticDB for MySQL is a cloud computing service designed for online analytical processing (OLAP). It can process large amounts of data in a highly concurrent and real-time manner. AnalyticDB for MySQL can be used in combination with MaxCompute to implement big data-driven business systems. You can use MaxCompute to calculate and mine data offline and generate high-quality data. Then, you can import the data to AnalyticDB for MySQL for business systems to perform analysis.

You can use one of the following methods to import data from MaxCompute to AnalyticDB for MySQL:

- Use the import and export feature of DMS for AnalyticDB for MySQL.
- Use DataWorks to configure a data synchronization task and configure the MaxCompute reader and AnalyticDB writer.

MaxCompute and Recommendation Engine

Recommendation Engine (RecEng) is a recommendation service framework established in the Alibaba Cloud computing environment. The recommendation service is typically composed of three parts: log collection, recommendation computing, and product connection. Both the input and output for offline recommendation computing are MaxCompute tables.

On the Resources page of the RecEng console, you can add a MaxCompute project as a RecEng computing resource in the same way as you add a cloud computing resource.

MaxCompute and Tablestore

Tablestore is a distributed NoSQL data storage service built on the Apsara distributed operating system of Alibaba Cloud. MaxCompute V2.0 allows you to directly access and process table data in Tablestore by using external tables.

MaxCompute and OSS

Object Storage Service (OSS) is a secure, cost-efficient, and highly reliable cloud storage service that can store large volumes of data. MaxCompute V2.0 allows you to directly access and process table data in OSS by using external tables.

MaxCompute and OpenSearch

OpenSearch is a large-scale distributed search engine platform developed by Alibaba Cloud. After data is processed by MaxCompute, you can add MaxCompute data sources to the OpenSearch platform to access MaxCompute data.

MaxCompute and Mobile Analytics

Mobile Analytics is a service launched by Alibaba Cloud to collect and analyze app usage data, which provides developers with an end-to-end digital operations service. When the basic analysis reports that come with Mobile Analytics cannot meet the personalized needs of app developers, app developers can synchronize data to MaxCompute with a few clicks. They can further process and analyze the data based on their business requirements.

MaxCompute and Log Service

Log Service allows you to quickly perform operations, such as data collection, consumption, delivery, query, and analysis. After data is collected, you need more personalized analysis and mining. You can synchronize Log Service data to MaxCompute by using Data Integration in DataWorks. Then, you can use MaxCompute to perform personalized and in-depth data analysis and mining on log data.

MaxCompute and RAM

Resource Access Management (RAM) is a service provided by Alibaba Cloud to manage user identities and resource access permissions.

Character sets supported by other Alibaba Cloud services

Service	Supported character set
Tablestore	UTF-8
PAI	UTF-8
OSS	UTF-8

Service	Supported character set
Quick BI	UTF-8
DataWorks	UTF-8, GBK, CP936, and ISO 8859 are supported when data is uploaded in DataStudio. However, all data is encoded in UTF-8 in DataWorks. UTF-8 and GBK are supported when data is downloaded.

24.3. Benefits

Excellent big data cloud service and real data sharing platform in China

- MaxCompute can be used for data warehousing, mining, analysis, and sharing.
- Alibaba Group uses this centralized data processing platform in several of its own services, such as Aliloan, Data Cube, DMP (Alimama), and Yu'e Bao.

Support for a large number of clusters, users, and concurrent jobs

- A single cluster can contain more than 10,000 servers and maintain 80% linear scalability.
- A single MaxCompute system supports more than 1 million servers in multiple clusters without limits. However, linear scalability is slightly affected. It also supports multi-data-center deployment in a zone.
- A single MaxCompute system supports more than 10,000 users, more than 1,000 projects, and more than 100 departments of multiple tenants.
- A single MaxCompute system supports more than 1 million jobs (daily submitted jobs on average) and more than 20,000 concurrent jobs.

Big data computing at your fingertips

You do not need to worry about the storage difficulties and prolonged computing processes caused by the increase of the data volume. MaxCompute automatically expands the storage and computing capabilities of clusters based on the data volume. This allows you to focus on data analysis and mining to maximize your data value.

Out-of-the-box service

You do not need to worry about the creation, configuration, and O&M of clusters. Only a few simple steps are required to upload data, analyze data, and obtain analysis results in MaxCompute.

Secure and reliable data storage

MaxCompute uses multi-level data storage and access control mechanisms to protect user data against loss, leaks, and interception. These mechanisms include multi-copy technology, read and write request authentication, and application and system sandboxes.

Reliable management nodes

MaxCompute uses the multi-node cluster architecture. The management nodes of each component feature high availability. The faults that occur on O&M management nodes do not interrupt your services.

Powerful fault tolerance

MaxCompute supports automatic fault tolerance for the failures of hard disks on servers in a cluster and supports hot swapping of hard disks. In the event of a hard disk failure, services can be restored within 2 minutes.

Comprehensive storage space management

MaxCompute allows you to query information about both the storage capacity and usage of distributed file systems. It enables you to manage data lifecycles. MaxCompute also allows you to store data in different locations based on the data value or tag. For example, you can write temporary files to SSDs to accelerate I/O operations. This allows you to use cluster data more efficiently. MaxCompute also supports the self-optimizing Zstandard compression algorithm that provides the optimal compression ratio.

Comprehensive data backup

- MaxCompute allows you to perform full or incremental data backup and restore data from storage media.
- MaxCompute allows you to back up data for clusters in different data centers. This meets the requirements of mutual data backups among multiple data centers. You can use Apsara Big Data Manager (ABM) to manage the backup process in a visualized manner.
- MaxCompute allows you to back up and restore the metadata, files, and tables of key components.

Secure and reliable access control

- MaxCompute allows you to manage data access permissions. The permissions include logon permissions, permissions to create tables, read and write permissions, and whitelist-related permissions.
- MaxCompute allows you to manage administrative permissions, including administrator classification, in the Apsara Uni-manager Management Console.
- MaxCompute allows you to manage user permissions in the Apsara Uni-manager Management Console in a centralized manner. You can view and manage the permission management features of all components in the system. You can also keep permission management details from common users and simplify permission management for administrators. This improves the usability and user experience of permission management.

Multi-tenancy for multi-user collaboration

MaxCompute allows you to configure data access policies. This way, you can enable multiple data analysts in an organization to collaborate and make data accessible to users who are granted the required permissions. This ensures data security and maximizes productivity.

- **Isolation:** You can submit the tasks of multiple tenants (projects) to different queues for concurrent running. Resources are isolated among tenants.
- **Permission:** You can manage different tenants in a centralized manner and dynamically configure, manage, and isolate tenant resources. You can also collect statistics on the usage of tenant resources and manage multi-level tenants.
- **Scheduling:** MaxCompute supports multi-tenant scheduling for multiple clusters and resource pools.

Multi-region deployment

- You can specify compute clusters to efficiently use computing resources.

- Data exchanges between clusters are completed within MaxCompute, and data replication and synchronization between clusters are managed based on the configured policies. Therefore, cross-region data processing is no longer involved, which significantly reduces the waiting time for data processing.

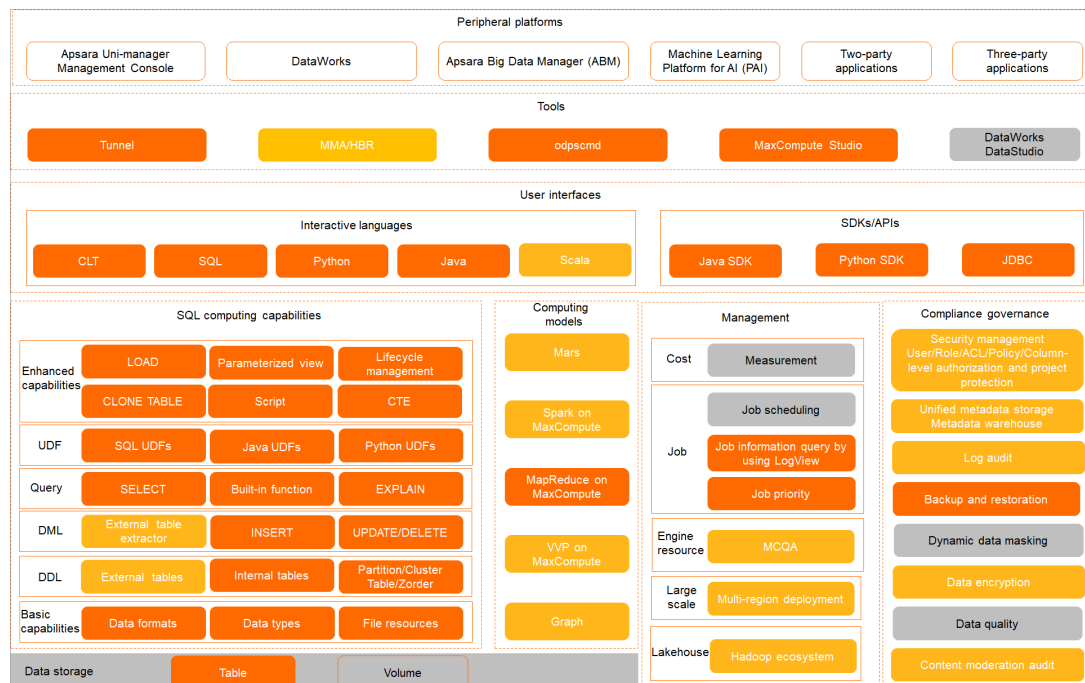
Multi-device support

You can use CPUs, hard disks, memory, and network interface controllers with different specifications in a single-component cluster to ensure maximum compatibility with existing devices. This applies only when cluster performance is not affected.

24.4. Architecture

This topic describes the architecture of MaxCompute. The architecture and descriptions are for reference only. They are subject to the released product type and supplementary features.

Architecture



■ indicates the basic features of MaxCompute. ■ indicates the enhanced features of MaxCompute. ■ indicates the features provided by external systems.

Category	Description

Category	Description
Peripheral platforms	MaxCompute supports the following peripheral platforms: • Apsara Uni-manager Management Console: a unified and intelligent O&M platform. For more information, see <i>Apsara Uni-manager Management Console User Guide</i>. • DataWorks: a visualization tool. You can use DataWorks to perform common operations, such as synchronize data, schedule jobs, and generate reports. For more information, see <i>DataWorks Technical White Paper</i>. • Apsara Big Data Manager (ABM): provides an easy method for field engineers to manage MaxCompute. For more information, see <i>Apsara Big Data Manager Technical White Paper</i>. • Machine Learning Platform for AI (PAI): a machine learning algorithm platform based on MaxCompute. For more information, see <i>Machine Learning Platform for AI Technical White Paper</i>. • Two-party applications: other Alibaba Cloud services supported by MaxCompute, such as DataV. • Three-party applications: other services that are compatible with MaxCompute.
Tools	MaxCompute supports the following tools: • Tunnel: a tunnel service. MaxCompute allows you to import heterogeneous data into or export the data from MaxCompute by using Tunnel. For more information, see <i>Tunnel</i> in <i>MaxCompute Product Introduction</i>. • MaxCompute Migration Assist (MMA): the data migration tool of MaxCompute. If you use MMA, Meta Carrier is used to access your Hive metastore service and capture Hive metadata. Then, MMA uses the Hive metadata to generate data definition language (DDL) statements and SQL statements of Hive user-defined table-valued functions (UDTFs). The DDL statements are used to create MaxCompute tables and their partitions. The SQL statements of Hive UDTFs are used to migrate data. • Hybrid backup recovery (HBR): integrates data backup and migration capabilities of Apsara Stack. • odpscmd: the MaxCompute client. For more information, see <i>Client</i> in <i>MaxCompute User Guide</i>. • MaxCompute Studio: the big data integrated development environment tool that is provided by MaxCompute. MaxCompute Studio is installed on a developer client. It is a development plug-in that Alibaba Cloud provides for the popular integrated development environment (IDE) IntelliJ IDEA. • DataWorks DataStudio: a visualized development platform provided by DataWorks. For more information, see <i>DataWorks User Guide</i>.

Category	Description
User interfaces	MaxCompute supports the following interfaces: - Interactive languages: CLI, SQL, Python, Java, and Scala. - SDKs and APIs: SDK for Java, SDK for Python, and Java Database Connectivity (JDBC). For more information, see <i>MaxCompute Developer Guide</i>.
SQL computing capabilities	MaxCompute supports the following SQL computing capabilities: - Enhanced capabilities: support LOAD, parameterized view, lifecycle management, and CLONE TABLE. - User-defined functions (UDFs): include SQL UDFs, Java UDFs, and Python UDFs. - Query: the query operations, such as SELECT and EXPLAIN statements and built-in functions. - Data manipulation language (DML) statements: include INSERT, UPDATE, and DELETE. - DDL statements: allow you to create internal tables, external tables, clustered tables, and partitioned tables. - Basic capabilities: support multiple data types and data formats and allow you to upload resource files. For more information, see <i>MaxCompute SQL</i> in <i>MaxCompute User Guide</i>.
Computing models	MaxCompute supports the following computing models: - Mars: a tensor-based unified distributed computing framework. Mars can use parallel and distributed computing technologies to accelerate data processing for Python data science stacks. For more information, see <i>Mars</i> in <i>MaxCompute User Guide</i>. - Spark on MaxCompute: a solution developed by Alibaba Cloud to enable the seamless use of Spark on the MaxCompute platform. It supplements a wide variety of features to MaxCompute. For more information, see <i>Spark on MaxCompute</i> in <i>MaxCompute User Guide</i>. - MapReduce on MaxCompute: allows you to run MapReduce jobs on MaxCompute. For more information, see <i>MaxCompute MapReduce</i> in <i>MaxCompute User Guide</i>. - VVP on MaxCompute: encapsulates the features of Realtime Compute for Apache Flink that is developed on the Ververica Platform (VVP) based on MaxCompute resources. You can use the Cupid joint computing platform to complete the operations related to real-time computing by using the underlying storage and computing resources of MaxCompute on the VVP UI. For more information, see <i>VVP On MaxCompute</i> in <i>MaxCompute User Guide</i>. - Graph: a processing framework designed for iterative graph computing. For more information, see <i>MaxCompute Graph</i> in <i>MaxCompute User Guide</i>.

Category	Description
Management	MaxCompute can be managed from the following aspects: • Cost: measures resource usage. • Job: provides mechanisms to manage jobs. For example, you can use these mechanisms to schedule jobs, use LogView to view job information, and set job priorities. • Engine resource: supports high-performance MaxCompute Query Acceleration (MCQA). • Large scale: allows you to deploy MaxCompute clusters across regions. • Lakehouse: a data management platform that combines data lakes and data warehouses. It integrates the flexibility and diverse ecosystems of data lakes with the enterprise-class deployment of data warehouses. For more information, see <i>MaxCompute Operations and Maintenance Guide</i>.
Compliance governance	MaxCompute allows you to use the following methods for compliance governance: • Security management: allows you to control the permissions of users and roles, and supports multiple authorization methods, such as ACL-based, policy-based, and column-level authorization. • Unified metadata storage: stores metadata in a centralized manner. • Log audit: audits different log data of different users. • Backup and restoration: allows you to back up and restore data from a storage system. • Dynamic data masking: allows you to query data masking rules in DataWorks. • Data encryption: uses Key Management Service (KMS) to encrypt data for storage. This way, MaxCompute can provide static data protection to meet the requirements of enterprise governance and security compliance. • Data quality: DataWorks provides an end-to-end platform that supports quality verification, notification, and management services for various heterogeneous data sources. • Content moderation audit: uses the content moderation engine to identify and audit pornographic, violent, and illegal content. For more information about security, see <i>MaxCompute Security White Paper</i>.
Data storage	MaxCompute stores data as tables or volumes.

24.5. Features

24.5.1. Tunnel

24.5.1.1. Terms

Tunnel is the data tunnel service provided by MaxCompute. You can use Tunnel to import data from various heterogeneous data sources into MaxCompute or export data from MaxCompute. As the unified channel for MaxCompute data transmission, Tunnel provides stable and high-throughput services.

Tunnel provides RESTful APIs and Java SDKs to facilitate programming. You can upload and download only table data (excluding view data) through Tunnel.

24.5.1.2. Tunnel features

- The channel through which data flows in to and out of MaxCompute
- Highly concurrent upload and download
- Horizontal expansion of service capabilities
- Tools based on MaxCompute Tunnel, such as TT, CDP, Flume, and Fluentd
- Support for reads and writes of tables (excluding views)
- Support for data writes in append mode
- Concurrency capabilities to improve total throughput
- Support for data upload only when target partitions exist
- Real-time upload mode

24.5.1.3. Data upload and download through Tunnel

Tunnel commands

```
odps@ > tunnel upload log.txt test_project.test_table/p1="b1",p2="b2 ";
```

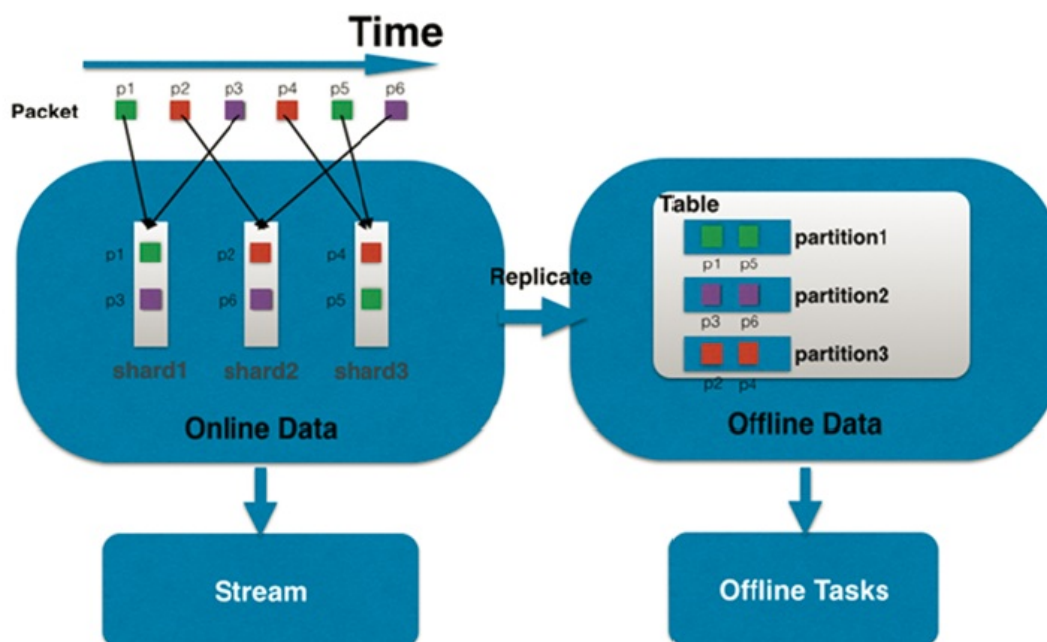
```
odps@ > tunnel download test_project.test_table/p1="b1",p2="b2" log.txt;
```

Notes

- Tunnel is a CLT based on the Tunnel SDK and can be used to upload local text files to MaxCompute or download data tables to your local device.
- You must create table partitions before using Tunnel.
- DataX, CDP, and TT provide enhanced Tunnel-based tools, which are used to exchange data between MaxCompute and relational databases.
- You can import log data by using the Flume and Fluentd tools.
- In some scenarios, you can develop custom tools based on Tunnel.

Real-time upload

- Upload in small batches
 - High QPS performance
 - Latency within milliseconds
 - Subscription available
- Real-time upload



24.5.2. SQL

24.5.2.1. Terms

The syntax of MaxCompute SQL is similar to SQL. It can be considered as a subset of standard SQL. However, MaxCompute SQL is not equivalent to a database, because it does not possess many characteristics that a database has, such as transactions, primary key constraints, and indexes. The maximum SQL statement size currently allowed in MaxCompute is 2 MB.

MaxCompute SQL offline computing is applicable to scenarios that have a large amount of data (measured in TBs) and that do not have high real-time processing requirements. It takes a relatively long time to prepare and submit each job. Therefore, MaxCompute SQL is not optimal for services that need to process thousands of transactions per second. MaxCompute SQL online computing is applicable to scenarios that require near-real-time processing.

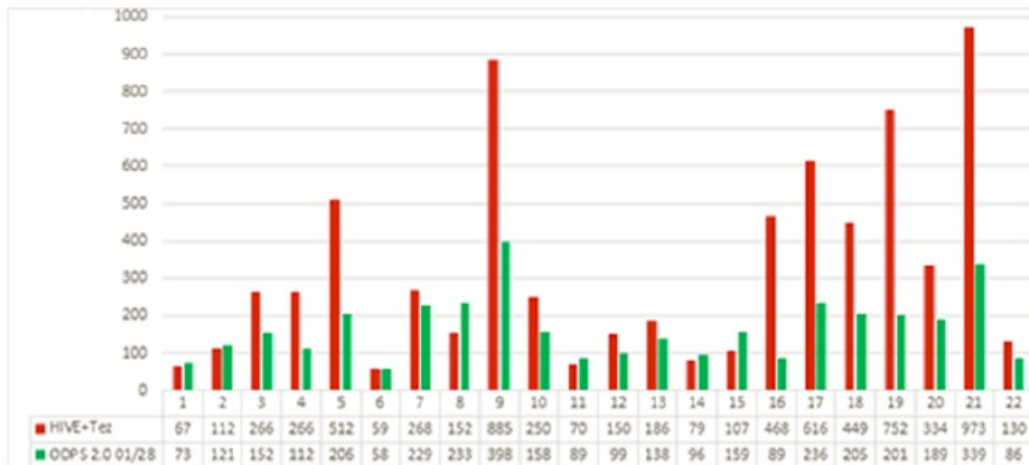
24.5.2.2. SQL characteristics

- It is suitable for processing large volumes of data (TBs or PBs).
- It has relatively high latency. The runtime of each SQL statement ranges from dozens of seconds to several hours.
- Its syntax is similar to that for Hive HQL. It is extended based on standard SQL syntax.
- It does not involve transactions or primary keys.
- It does not support UPDATE and DELETE operations.

24.5.2.3. Comparison with open source products

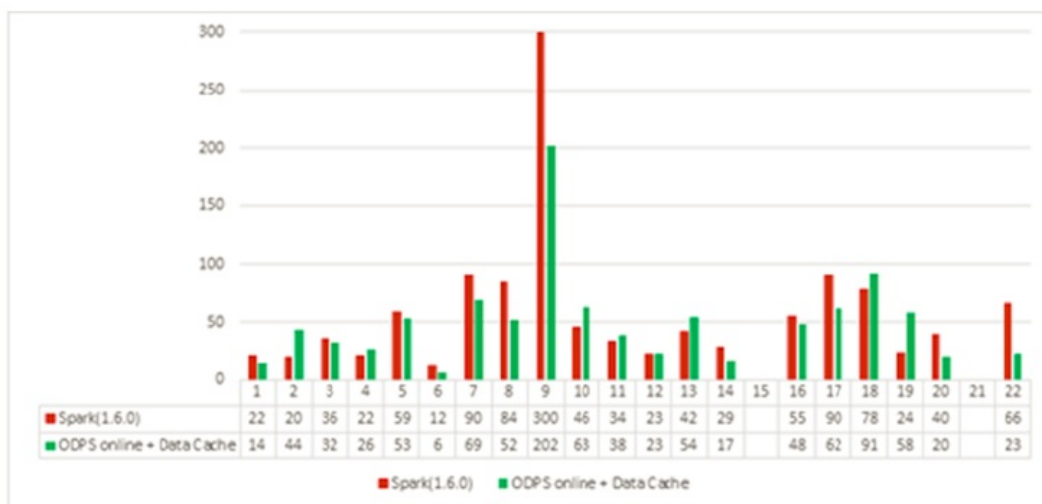
- TPC-H 1 TB data benchmark: Compared with Hive (Apache Hive-1.2.1-bin + Tez-UI-0.7.0 with CBO), MaxCompute has a 95.6% improvement in performance.

MaxCompute 2.0 VS Hive



- TPC-H 450 GB data benchmark: Compared with Spark SQL V1.6.0 (the latest release), MaxCompute has a 17.8% improvement in performance.

MaxCompute 2.0 VS Spark SQL



24.5.3. MapReduce

24.5.3.1. Terms

MapReduce is a programming model, which is basically equivalent to Hadoop MapReduce. The model is used for parallel MaxCompute operations on large-scale data sets (measured in TBs).

MaxCompute provides a MapReduce programming interface. You can use Java APIs, which is provided by MapReduce, to write MapReduce programs for processing data in MaxCompute.

Note All data in MaxCompute is stored as tables. The inputs and outputs of MaxCompute MapReduce can only be tables. Custom output formats are not supported, and no interface, such as a file system, is provided.

24.5.3.2. MapReduce characteristics

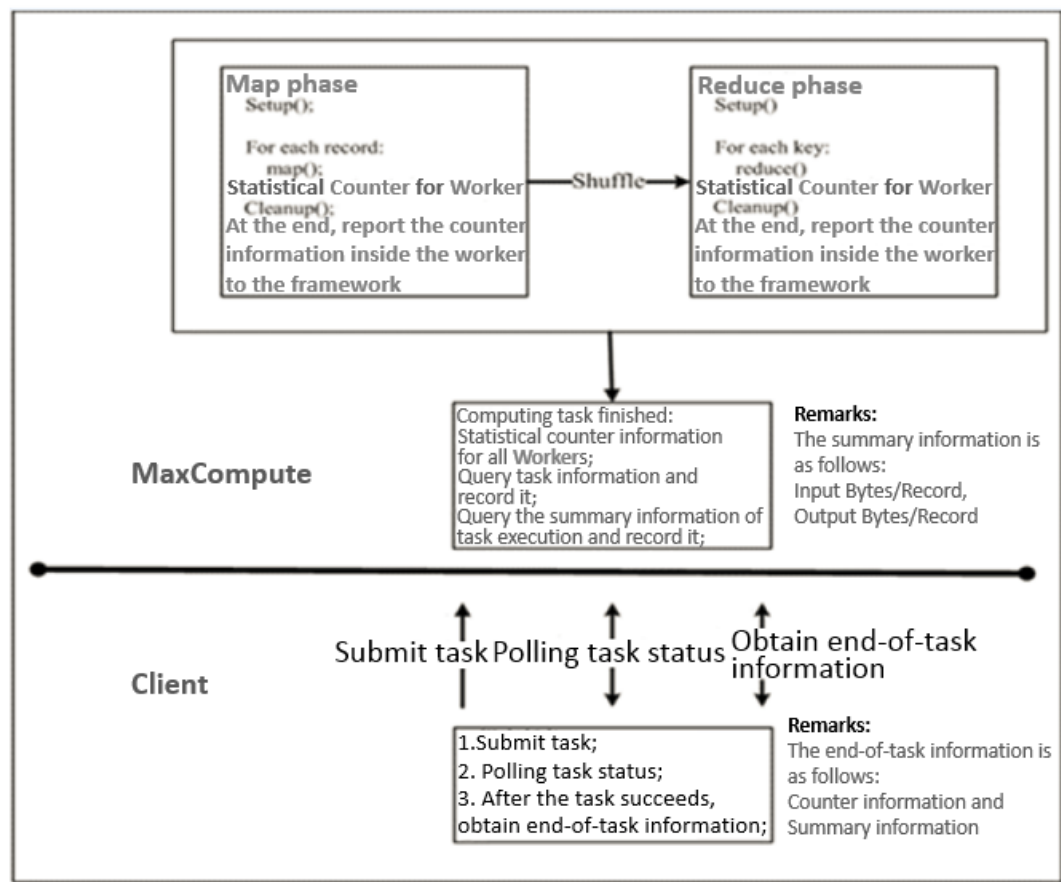
- It only supports the input and output of MaxCompute built-in data types.

- It supports the input and output of multiple tables to different partitions.
- It reads resources.
- It does not support using views as data inputs.
- It supports MapReduce programming only in the JDK 1.8 environment.
- It provides a limited sandbox security environment.

24.5.3.3. MaxCompute MapReduce process

The following figure shows the MapReduce process in MaxCompute:

MapReduce process



24.5.3.4. Hadoop MapReduce VS MaxCompute

MapReduce

The following table describes the comparison between Hadoop MapReduce and MaxCompute MapReduce.

Mapper/Reducer

Mapper/Reducer	
Hadoop MapReduce	MaxCompute MapReduce

Map (InKey key, InputValue value, OutputCollector<OutKey, OutValue> output, Reporter reporter)	Map (long key, Record record, TaskContext context)
Reduce (InKey key, Iterator<InValue> values, OutputCollector<OutKey, OutValue> output, Reporter reporter)	Reduce (IRecord key, Iterator<Record> values, TaskContext context)

MapReduce

```
@Override
public void map(long recordNum, Record record, TaskContext context)
    throws IOException {
    for (int i = 0; i < record.getColumnCount(); i++) {
        word.set(new Object[] { record.get(i).toString() });
        context.write(word, one);
    }
}
```

24.5.4. Graph

24.5.4.1. Terms

Graph is the computing framework of MaxCompute designed for iterative graph processing. It provides programming interfaces similar to Pregel, allowing you to use Java SDKs to develop efficient machine learning and data mining algorithms.

Graph jobs use graphs to build models. This process outputs a result after performing iterative graph editing and evolution.

24.5.4.2. Graph characteristics

- It is a graphic computing programming model (similar to Google Pregel).
- It loads data to the memory, which is superior in multiple iteration scenarios.
- It can be used to develop machine learning algorithms.
- It can support 10 billion vertices and 150 billion edges.
- Its typical applications include:
 - PageRank
 - K-means clustering
 - Level 1 and level 2 relationships and shortest path
- Graph jobs process graph data.
- The original data is stored in tables. The user-defined Graph Loader loads data in the table as vertexes and edges.
- It supports iterative computing.

24.5.4.3. Graph relational network models

A relational network engine provides a variety of business-oriented relational network models. It helps you quickly implement relational data mining at finer granularities.

Community discovery

- Input to the engine: relational data.
- Engine output: IDs and community IDs.
- Computing logic: locates N communities with the optimal global network connection. The communities are close enough internally, and sparse enough in between.

Semi-supervised category

- Input to the engine: problematic IDs.
- Engine output: potentially problematic IDs and weights.
- Computing logic: uses existing problematic IDs (of one or more categories) to determine potential problematic IDs of the same or multiple categories and corresponding weights based on the entire network connection relationships.

Isolated point detection

- Input to the engine: relational data.
- Engine output: isolated points and weights.
- Computing logic: determines whether there are relatively isolated nodes using the connection relationships in a relational network, and generates the result.

Key point mining

- Input to the engine: relational data.
- Engine output: key point IDs and categories.
- Computing logic: calculates the key type nodes in a computing network using the connection relationships (such as centrality, influence, and betweenness centrality) in a relational network.

Level N relationships

- Input to the engine: relational data.
- Engine output: retrievable relational networks.
- Computing logic: manages multi-dimensional relationships using the connection relationships in the relational network, and creates indexes to facilitate the query for specific associations of an ID.

24.5.5. Unstructured data processing in integrated computing scenarios

Alibaba Cloud introduced the MaxCompute-based unstructured data processing framework so that MaxCompute SQL can directly process external user data, such as unstructured data from Object Storage Service (OSS). You are no longer required to first import data into MaxCompute tables.

You can execute a DDL statement to create an external table in MaxCompute and associate the table with external data sources. This table can then act as an interface between MaxCompute and external data sources. External tables can be accessed in the same way as standard MaxCompute tables. You can fully use the computing capabilities of MaxCompute SQL to process external data.

MaxCompute allows you to create external tables to process data from the following data sources:

- Internal data sources: OSS, Tablestore, AnalyticDB, ApsaraDB RDS, Alibaba Cloud HDFS, and TDDL
- External data sources: open source HDFS, MongoDB, and HBase

24.5.6. Unstructured data processing in

MaxCompute

MaxCompute has the following problems when processing unstructured data: MaxCompute stores data as volumes and must export generated unstructured data to an external system for processing.

To alleviate these problems, MaxCompute uses external tables to enable connections between MaxCompute and various data types. MaxCompute uses external tables to read and write data volumes as well as process unstructured data from external sources such as OSS.

24.5.7. Enhanced features

24.5.7.1. Spark on MaxCompute

24.5.7.1.1. Terms

Spark on MaxCompute is a solution developed by Alibaba Cloud to enable seamless use of Spark on the MaxCompute platform, extending the functions of MaxCompute.

Spark on MaxCompute provides a native Spark user experience with its native Spark components and APIs. It allows access to MaxCompute data sources and better security for multi-tenant scenarios. It also offers a management platform enabling Spark jobs to share resources, storage, and user systems with MaxCompute jobs. This guarantees high performance and low costs. Spark can work with MaxCompute to create better and more efficient data processing solutions. Spark Community applications can run seamlessly in **Spark on MaxCompute**.

Spark on MaxCompute has an independent data development node in DataWorks and supports data development in DataWorks.

24.5.7.1.2. Features of Spark on MaxCompute

Processing of data from MaxCompute and unstructured data sources

- Processes MaxCompute tables through APIs based on Scala, Python, Java, and R programming languages.
- Processes MaxCompute tables through components such as Spark SQL, Spark MLlib, GraphX, and Spark Streaming.
- Can process unstructured data from Alibaba Cloud OSS.

User-friendly experience and management functions

- Supports job submission in a way similar to Spark on YARN. **Spark on MaxCompute** is compatible with YARN and HDFS APIs.
- Supports components including Spark SQL, Spark MLlib, GraphX, and Spark Streaming.
- Can work with SQL and Graph components of MaxCompute to form optimized solutions.
- Can connect to the native Spark UI.
- Allows you to directly use the powerful management functions of MaxCompute.

- Supports not only Spark Community but also tools such as client, Livy, and Hue.

Scalability

Spark and MaxCompute share cluster resources. Spark resources can be scaled from large-scale MaxCompute clusters.

24.5.7.1.3. Spark features

The following table describes **Spark on MaxCompute** features.

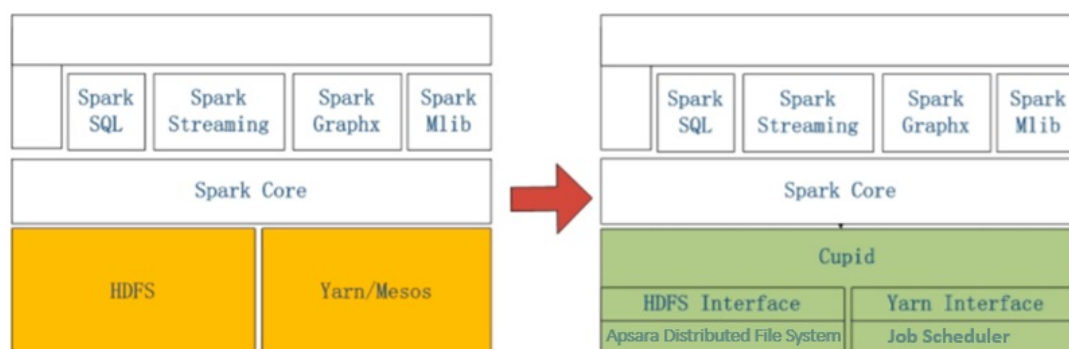
Features

Type	Feature	Description
Distributed cluster	Cluster deployment Cluster monitoring	Provide an O&M platform to monitor clusters and nodes.
Data processing component	Support for components such as Spark SQL, Spark MLlib, GraphX, and Spark Streaming	Provide native Spark components.
Job management	Centralized resource management, life cycle management, and authentication	The features are available through compatible YARN APIs.
Data sources	Unstructured data Table data sources in MaxCompute	Provide data processing capabilities of SQL and MapReduce on MaxCompute.
Security management	User identification, data authentication, and multi-tenant job isolation	Harden Spark security through authentication and sandboxes.

24.5.7.1.4. Spark architecture

The following figure shows the architectural comparison between **Spark on MaxCompute** and **native Spark**.

Architectural comparison between Spark on MaxCompute and native Spark



 **Note** On the left is the native Spark architecture and on the right is the Spark on MaxCompute architecture.

As shown in the figure, Spark on MaxCompute has the computing capabilities of native Spark and the functions related to management, O&M, scheduling, security, and data interconnection. The management function of Spark is implemented by starting a Cupid Task instance of MaxCompute. The resource application function is realized through layer-1 YARN APIs provided by MaxCompute. The security function is offered through the sandbox mechanism of MaxCompute. The processing of and interconnection between data and metadata are also made available. The module details are described as follows:

- The MaxCompute control cluster starts a Spark driver by using the Cupid Task instance. The Spark driver uses YARN APIs to apply for resources from FuxiMaster, the central resource manager.
- The MaxCompute control cluster manages user quota consumed by running Spark instances, life cycles of Spark instances, and permissions on accessible data sources.
- The MaxCompute computing cluster starts a Spark driver and Executor as parent and child processes and executes Spark code in the sandbox of MaxCompute, ensuring security in multi-tenant scenarios.
- MaxCompute allows you to use the native Spark UI through its Proxy Server and manage job information through its management components.

24.5.7.1.5. Benefits of Spark on MaxCompute

Support for the complete Spark ecosystem

Provides consistent user experience with that of open source Spark.

Full integration with MaxCompute

Implements centralized management of resources, data, and security features for both Spark and MaxCompute.

Combination of Spark and the Apsara system

Combines the flexibility and ease of use of Spark with the high availability, scalability, and stability of the Apsara system.

Support for multi-tenancy

Reduces costs by centrally scheduling resources in large-scale clusters and ensuring high performance of physical machines.

Support for cross-cluster scheduling

Maximizes the efficiency of cluster resources by effectively allocating clusters and scheduling resources across clusters.

Support for real-time scaling of Spark resources

Scales resources in Spark Community in real time to better utilize resources and avoid waste.

 **Note** Real-time Spark resource scaling is not enabled on all MaxCompute clusters. To use this function, contact the MaxCompute team.

24.5.7.2. Elasticsearch on MaxCompute

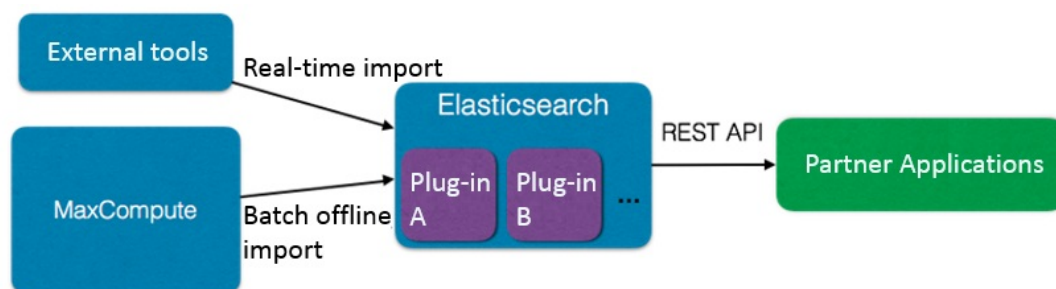
24.5.7.2.1. Overview

Elasticsearch on MaxCompute is an enterprise-class full-text retrieval system developed by Alibaba Cloud to retrieve large volumes of data with near-real-time search performance.

Elasticsearch on MaxCompute provides elastic full-text retrieval and supports native Elasticsearch APIs. You can import data from heterogeneous data sources and perform O&M for clusters and services. The centralized scheduling and management capabilities of MaxCompute allow Elasticsearch to provide more efficient core services for data retrieval at large volumes. Elasticsearch on MaxCompute can also work with plug-ins available from the Elasticsearch open source community to enhance retrieval functions.

Elasticsearch on MaxCompute allows you to use tools to import data from external sources in real time. You can also import offline data from MaxCompute. After the imported data is indexed, Elasticsearch on MaxCompute provides retrieval services through RESTful APIs. The following figure shows its usage.

Elasticsearch on MaxCompute usage



24.5.7.2.2. Features of Elasticsearch on MaxCompute

Distributed cluster architecture

- Improves retrieval and reliability of data with a distributed architecture.
- Supports elastic scaling.
- Supports dynamic scaling.
- Supports service-level O&M and monitoring.

Robust full-text retrieval

- Performs full-text retrieval at the word, phrase, sentence, and section levels.
- Available in languages such as Chinese and English.
- Provides precise word segmentation with 100% recall for Chinese information retrieval.
- Supports complex searching methods, such as Boolean retrieval, proximity search, and fuzzy search.
- Sorts search results by relevance, field, and custom weight, and allows for secondary sorting.
- Performs statistical classification and analysis of search results.
- Allows real-time indexing and retrieval, so that inserted data can be retrieved immediately.
- Allows an index to be used multiple times after it is created.
- Allows you to modify the index structure in real time or rebuild the index to re-distribute data.

Support for multiple data sources

- Imports data from native Elasticsearch interfaces.
- Provides data import tools for MaxCompute.
- Supports full and incremental update.

Reliability

- Stores data in multiple copies, preventing user data from being lost during the downtime of machines.
- Implements a high availability architecture and comprehensive failover for nodes and services.
- Provides comprehensive O&M and monitoring functions.
- Authenticates access to protect data from malicious operations and ensure security.

24.5.7.2.3. Elasticsearch features

Elasticsearch on MaxCompute features are described as follows:

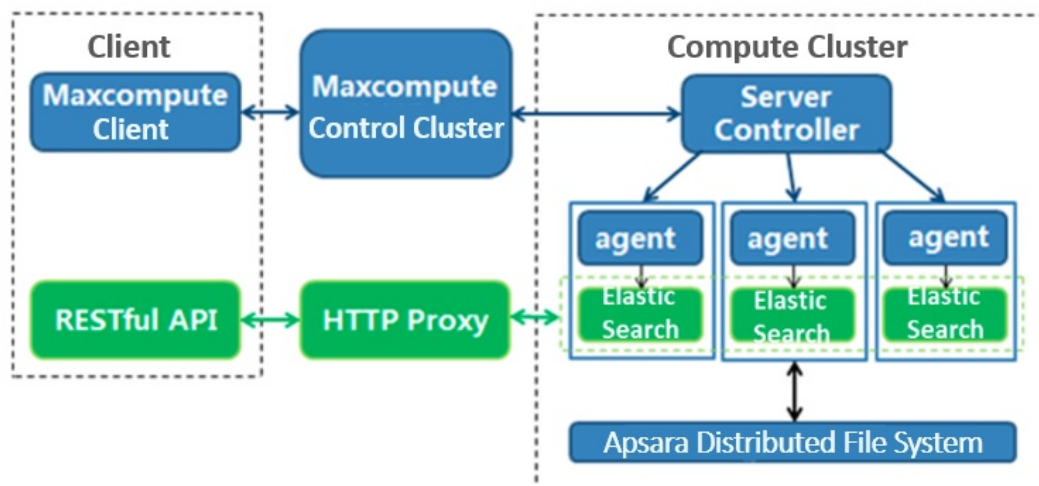
Features

Type	Feature	Description
Distributed cluster	Cluster deployment Cluster monitoring	Provide an O&M platform to monitor clusters, nodes, and indexes.
Retrieval management	Index configuration management Structure definition and index rebuilding	Provide a retrieval management platform and support configuration.
Full-text retrieval	Retrieval Sorting Statistical analysis	The features are provided through RESTful APIs.
Data collection	Elasticsearch data import APIs MaxCompute data import tools Full and incremental collection	Support a variety of interfaces to collect native data. Provide integrated tools to import MaxCompute data.
Service authentication	Service-level user authentication	Allow you to configure user authentication in a centralized manner.

24.5.7.2.4. Elasticsearch architecture

Elasticsearch on MaxCompute provides core search engine services, management platforms for O&M and indexes, MaxCompute management system, and MaxCompute data import tools. It can work with universal data import interfaces and data retrieval SDKs of Elasticsearch, enabling you to retrieve applications and perform full-text retrieval of large volumes of data. The following figure shows the overall architecture.

Overall architecture



An Elasticsearch cluster corresponds to a MaxCompute Server Task instance in MaxCompute. You can quickly and flexibly deploy, operate, and expand Elasticsearch clusters on a MaxCompute client. In the overall architecture,

- The MaxCompute control cluster starts Server Controller and forwards control requests from a client.
- Server Controller is the core component for Elasticsearch cluster management. It applies for resources, starts each Elasticsearch node, and responds to the control requests that a client forwards through the control cluster. It also returns the running status of Elasticsearch clusters or adjusts the clusters.
- An agent starts Elasticsearch node processes, monitors node running status, handles failover events, and executes tasks distributed by Server Controller.
- Elasticsearch on MaxCompute stores its data in Apsara Distributed File System. Once a node is started successfully, Elasticsearch on MaxCompute can provide services through HTTP Proxy and allow users to use its functions through RESTful APIs.

24.5.7.2.5. Benefits

Integration of big data computing and data retrieval for resource sharing

Elasticsearch on MaxCompute can access and import MaxCompute data to an Elasticsearch cluster to perform a full-text search. This facilitates centralized data management and usage.

Centralized management of computing and storage resources

You do not need to worry about the storage problems and prolonged computing tasks caused by the increase of the data volume. Elasticsearch on MaxCompute supports automatic scaling of your cluster storage and retrieval capacities based on the volume of your data. This way, you can focus on data analytics and mining to maximize your data value.

Provision of services such as Elasticsearch cluster deployment and O&M

You do not need to worry about cluster creation, configuration, and O&M. Only a few simple steps are required to upload data, analyze data, and obtain analysis results in the offline analysis service.

Secure and reliable data storage

Elasticsearch on MaxCompute uses the multi-replica technology to store user data at multiple layers. This prevents the loss, leak, and interception of data.

Open service interfaces

Elasticsearch on MaxCompute provides Elasticsearch SDKs that are native and open. This allows you to import, index, and retrieve data by using **Elasticsearch on MaxCompute**.

Multi-tenancy for multi-user collaboration

- **Isolation:** Elasticsearch on MaxCompute supports the cross-cluster search feature and allows you to submit tasks to different clusters for execution. Resources among clusters are isolated. Elasticsearch on MaxCompute does not support single-cluster multi-tenancy.
- **Permissions:** You can manage clusters in a centralized manner to implement the configuration, management, isolation, and usage statistics of cluster resources. In addition, it supports multi-level permissions and multi-level tenant management based on Apsara Stack.
- **Scheduling:** Elasticsearch on MaxCompute supports multi-tenant scheduling for multiple clusters and multiple resource pools.

24.5.8. Multi-region deployment

24.5.8.1. MaxCompute multi-region deployment

This topic describes the multi-region deployment supported by MaxCompute. Control clusters are deployed in a centralized manner and used to configure resources and manage computing tasks. Compute clusters are separately deployed in each region to create projects and distribute computing tasks.

The multi-region deployment of MaxCompute has the following features:

- A MaxCompute system can manage clusters in different regions.
- Data exchanges between clusters are implemented within MaxCompute, and data replication and synchronization between clusters are managed based on configured policies.
- Metadata is stored in a centralized manner. Therefore, the infrastructure requirements, such as the network connections of different data centers, are relatively high.
- A unified account system is used.
- The development systems for big data applications, such as DataWorks, are used for clusters in all regions.
- MaxCompute must run in multi-cluster mode to support multi-region deployment.

 **Note** Take note of the following conditions and limits on changes to the multi-cluster mode:

- The network bandwidth must be sufficient to support multi-region data synchronization and link redundancy.
- Control clusters in the central region have a high latency for basic services, such as Apsara Stack DNS and Tablestore. Therefore, we recommend that you deploy basic services in the same data center to ensure that the network latency remains within 5 ms.
- The network latency between control clusters in the central region and compute clusters in other regions must be within 20 ms.
- Clocks must be synchronized between clusters in different regions and between servers in the same cluster.
- The network bandwidth must be sufficient to support data replication between clusters.
- Apsara Stack DNS is required.
- Servers in different clusters can communicate with each other, and the clusters have similar network infrastructure (1-Gigabit or 10-Gigabit).

- The O&M and upgrades for multi-region deployment are different from those for single-cluster deployment. Multi-region deployment requires higher on-site O&M capabilities.
- MaxCompute supports cross-region multi-cluster (sub data centers) distributed computing. It uses the global job scheduling feature of the primary data center to balance the resource usage among clusters. It schedules jobs to the most appropriate cluster based on cluster information, such as the default settings, historical analysis, data distribution, and cluster load. Then, it executes the jobs and generates query results. MaxCompute supports history- and cost-based optimization policies of SQL queries. Remote clusters in unified global data management mode allow you to uniformly schedule and manage resources that belong to multiple clusters in different data centers.

24.6. Scenarios

24.6.1. Scenario 1: Migrate data to the cloud cost-effectively and quickly

Usage scenario: The customer is a data and information service provider focusing on the new energy power sector. The customer's target is to build a cloud platform for Internet big data application services of the new energy industry.

Results: The customer's entire business system has been migrated to the cloud within three months. The data processing time is decreased to less than one third when compared with the customer-built system. Cloud data security is ensured through multiple security mechanisms.

Customer benefits:

- **More focus on its core business:** The entire business system is migrated to the cloud within three months, which enables the customer to use a variety of cloud resources to improve the business.
- **Low investment and O&M costs:** The cloud platform helps to significantly lower the costs of infrastructure construction, O&M personnel, and R&D when compared with a customer-built big data platform.
- **Security and stability:** Alibaba Cloud's comprehensive service and stable performance guarantee data security on the cloud.

24.6.2. Scenario 2: Improve development efficiency and reduce storage and computing costs

Usage scenario: Massive log analysis services for weather query and advertising business are provided to meet the business needs of an emerging mobile Internet company aiming for an excellent weather service provider.

Results: After the Internet company's log analysis business is migrated to MaxCompute, the development efficiency is improved by more than five times, the storage and computing costs are reduced by 70%, and 2 TB of log data is processed and analyzed every day. This more efficiently empowers its personalized marketing strategies.

Customer benefits:

- **Improved work efficiency:** All log data is analyzed by using SQL, and the work efficiency is increased by more than 5 times.
- **Improved storage usage:** The overall storage and computing cost is reduced by 70%, and the performance and stability are also improved.
- **Personalized service:** Machine learning algorithms on MaxCompute are used to perform in-depth data mining and provide personalized services for users.
- **Easy use of big data:** MaxCompute provides plugins for a variety of open-source software to easily migrate data to the cloud.

24.6.3. Scenario 3: Use mass data to achieve precision marketing for millions of users

Usage scenario: To meet the business needs of a community-oriented vertical e-commerce app that focuses on the manicure industry, you can use MaxCompute to build a big data platform for the app. It is mainly used in four aspects: business monitoring, business analysis, precision marketing, and recommendation.

Results: This e-commerce app uses the big data platform built based on MaxCompute to achieve precision marketing for millions of users through the computing capability of MaxCompute, making e-commerce business more agile, intelligent, and insightful. The platform can quickly respond to the data and analysis needs of new business.

Customer benefits:

- **Improved business insights:** Through the computing capabilities of MaxCompute, precision marketing for millions of users is achieved.
- **Data-driven business:** The platform improves the business data analysis capability and effectively monitors business data to better empower businesses.
- **Fast response to business needs:** The MaxCompute ecosystem can quickly respond to changing business data analysis needs.

24.6.4. Scenario 4: Achieve precision marketing with big data

Usage scenario: MaxCompute is used to meet the business needs of an Internet company that focuses on precision marketing and advertising technologies and services. A core big data-based precision marketing platform will be built for the company.

Results: Based on MaxCompute, the company builds a core big data-based precision marketing platform. All log data is stored in MaxCompute, and offline scheduling and analysis are performed through DataWorks.

Customer benefits:

- **Efficient and low-cost analysis of massive data:** Statistical analysis of massive data can reduce expenditures by half to meet the same business needs, effectively saving costs and helping startup enterprises grow rapidly.
- **Real-time data query and analysis:** MaxCompute helps the enterprise establish technical advantages, overcoming the technical bottleneck of massive data processing and analysis, and real-time query and analysis. MaxCompute collects, analyzes, and stores more than 2 billion visitor activities every day. At the same time, it performs millisecond-level queries in hundreds of millions of log tables based on user requirements.
- **Machine learning platform with low entry barrier:** As for a precision marketing and advertising provider, the quality of algorithm models is directly linked to its final revenue. Therefore, selecting the ease-of-use MaxCompute machine learning platform with low entry barrier can get twice the result with half the effort.

24.7. Limits

None.

24.8. Terms

project

The basic unit of operation in MaxCompute. A MaxCompute project is similar to a database or schema in a traditional database. MaxCompute projects set boundaries for isolation and access control between different users. A user can have permissions on multiple projects.

 **Note** After being authorized, a user can access objects within a project, such as tables, resources, functions, and instances, from other projects.

table

The data storage unit in MaxCompute. A table is a two-dimensional data structure composed of rows and columns. Each row represents a record, and each column represents a field of the same data type. One record can contain one or more columns. The column names and data types comprise the schema of a table.

 **Note** There are two types of MaxCompute tables: external tables and internal tables.

partitioned table

A logical structure used to divide a large table into smaller pieces called partitions. You can specify a partition when creating a table. Specifically, several fields in the table can be specified as partition columns. If you specify the name of a partition you want to access, the system only reads data from the specified partition instead of scanning the entire table, thus reducing costs and improving efficiency.

lifecycle

The validity period of a MaxCompute table or partition. The lifecycle of a MaxCompute table or partition is measured from the last update time. If the table or partition has not undergone any changes within a specified amount of time, MaxCompute will automatically recycle it. This amount of time is specified by the lifecycle.

- Lifecycle unit: days, positive integers only.
- When a lifecycle is specified for a non-partitioned table, the lifecycle is counted from the last time the table data was modified (LastDataModifiedTime). If the table has not been modified before the end of the lifecycle, MaxCompute will automatically recycle the table in a manner similar to the DROP TABLE operation.
- When a lifecycle is specified for a partitioned table, you can decide whether a partition should be recycled based on the LastDataModifiedTime value of the partition. Unlike non-partitioned tables, a partitioned table will not be deleted even if its last partition has been recycled.

 **Note** Lifecycle scanning is performed at a scheduled time each day, and entire partitions are scanned. If a partition has not undergone any changes within its lifecycle, MaxCompute will automatically recycle it. Assume that the lifecycle of a partitioned table is one day and that the partition data was last modified at 15:00 on the 17th. If the table is scanned before 15:00 on the 18th, the aforementioned partition will not be recycled. During the lifecycle scanning scheduled on the 19th, if the last modification time of the partition exceeds the lifecycle period, the partition will be recycled.

- You can configure a lifecycle for tables, but not for partitions. You can specify a lifecycle when creating a table.
- If no lifecycle is specified, the table or partition cannot be automatically recycled by MaxCompute.

data type

A property of a field that defines the kinds of data the field can store. Columns in MaxCompute tables must be of one of the following data types: TINYINT, SMALLINT, INT, BIGINT, STRING, FLOAT, BOOLEAN, DOUBLE, DATETIME, DECIMAL, VARCHAR, BINARY, TIMESTAMP, ARRAY, MAP, and STRUCT.

resource

A unique concept in MaxCompute. To accomplish tasks by using user-defined functions (UDFs) or MapReduce features in MaxCompute, you must use resources.

 **Note** Resource types in MaxCompute include file, MaxCompute table, JAR (compiled JAR package), and archive. Compressed files are identified by the extensions of resource names. Supported file types include .zip, .tgz, .tar.gz, .tar, and .jar.

function

A piece of code that operates as a single logical unit. MaxCompute provides SQL computing capabilities. In MaxCompute SQL, you can use built-in functions for computing and calculation. When the built-in functions are not sufficient to meet your requirements, you can use the Java programming interface provided by MaxCompute to develop UDFs.

 **Note** UDFs can be further divided into scalar-valued functions, user-defined aggregate functions (UDAFs), and user-defined table functions (UDTFs).

task

The basic computing unit of MaxCompute. Computing jobs such as those involving SQL and MapReduce functions are completed by using tasks.

task instance

A snapshot of a task taken at a specified time. In MaxCompute, some tasks are converted into instances when being executed and subsequently exist as MaxCompute instances.

resource quota

A per-process limit on the use of system resources. There are two types of quotas: storage and computing. MaxCompute allows you to set an upper limit of storage for a project. When the storage space occupied approaches the upper limit, MaxCompute triggers an alert. The computing quota limits the use of memory and CPU resources. The memory usage and CPU utilization of running processes in a project cannot exceed the specified upper limit.

ACID semantics

This topic describes the ACID semantics of MaxCompute for concurrent jobs. ACID is an acronym that stands for Atomicity, Consistency, Isolation, Durability.

Terms

- Operation: a single job submitted in MaxCompute.
- Data object: an object that contains data, such as a non-partitioned table or partition.
- INTO jobs: such as INSERT INTO and DYNAMIC INSERT INTO.
- OVERWRITE jobs: such as INSERT OVERWRITE and DYNAMIC INSERT OVERWRITE.
- Data upload with Tunnel: an INTO or OVERWRITE job.

Description of ACID semantics

- Atomicity: An operation is either fully completed or not executed at all.
- Consistency: The integrity of data objects is not compromised during the entire period of an operation.
- Isolation: An operation is completed independent of other concurrent operations.
- Durability: After an operation is complete, data is available in its current state even in the event of a system failure.

Description of ACID semantics in MaxCompute

- Atomicity
 - At any time, MaxCompute ensures that only one job succeeds in the case of a conflict, and all other conflicting jobs fail.

- The atomicity of the CREATE, OVERWRITE, and DROP operations on a single table or partition can be guaranteed.
- Atomicity is not supported in cross-table operations such as MULTI-INSERT.
- In extreme cases, the following operations may not be atomic:
 - The DYNAMIC INSERT OVERWRITE operation is performed on more than 10,000 partitions.
 - INTO operations fail because the data cleansing fails during transaction rollback. This does not cause raw data loss.
- Consistency
 - OVERWRITE jobs ensure consistency.
 - If an OVERWRITE job fails due to a conflict, data from the failed job may remain.
- Isolation
 - Non-INTO operations ensure that read operations are committed.
 - INTO operations can be performed in scenarios where read operations are not committed.
- Durability

MaxCompute ensures data durability.

24.9. Storage performance

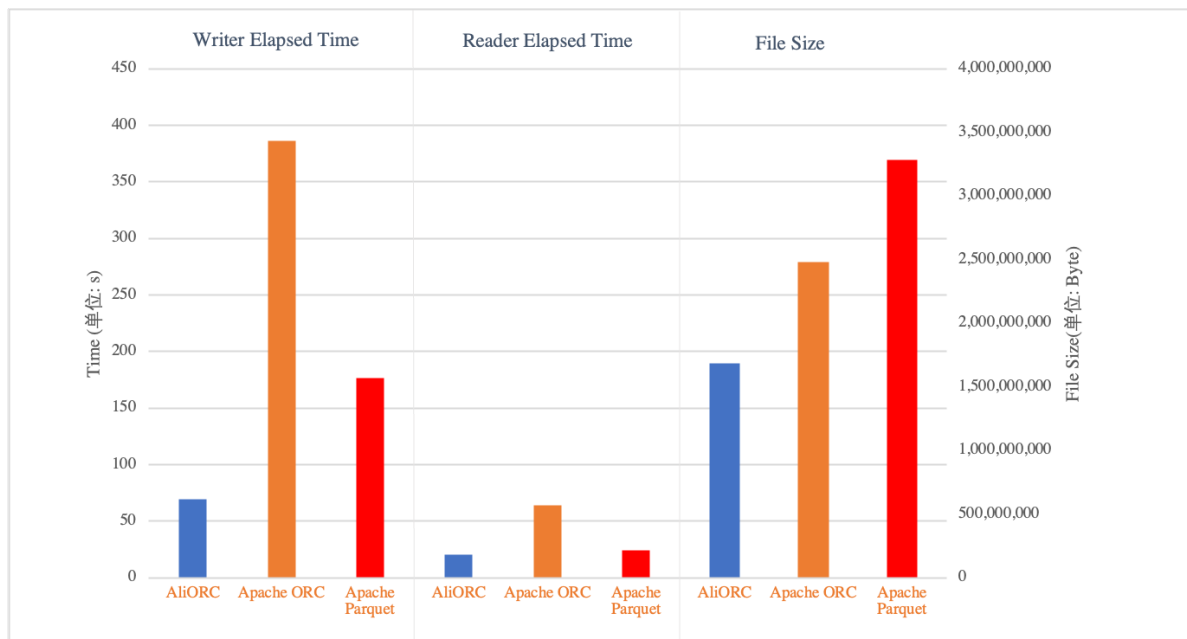
The data storage format of MaxCompute has been updated to AliORC since February 2020. This topic compares AliORC with Apache Optimized Row Columnar (ORC) and Apache Parquet and their compression ratios based on TPC Benchmark DS (TPC-DS) tests. The comparison helps you better understand the data storage performance of MaxCompute.

Test results

- The following table describes the comparison between AliORC and Apache ORC and between AliORC and Apache Parquet. The comparison is based on the dataset that contains 24 test tables.

Storage format	Duration of a data write (writer elapsed time)	Duration of a data read (reader elapsed time)	Volume of data stored (file size)
AliORC and Apache ORC	AliORC reduces the duration by more than 85%.	AliORC reduces the duration by more than 76%.	AliORC reduces the data volume by more than 8%.
AliORC and Apache Parquet	AliORC reduces the duration by more than 50%.	AliORC reduces the duration by more than 28%.	AliORC reduces the data volume by more than 22%.

The following figure shows the test results of the dataset.

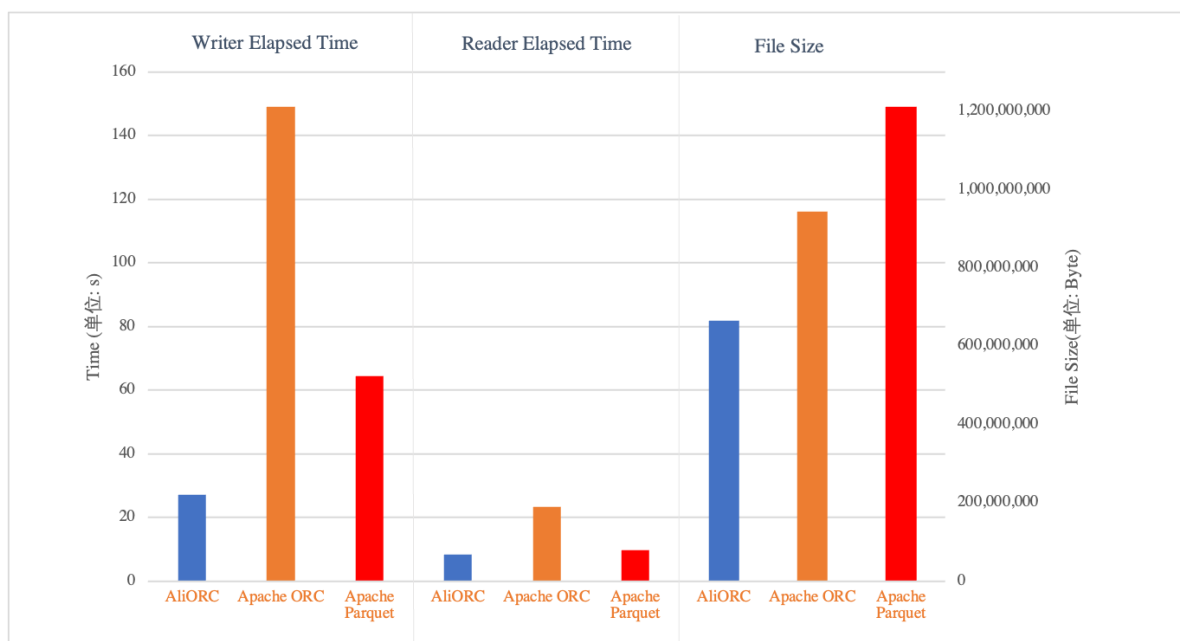


Parameters:

- Writer elapsed time: the time required to import CSV data of TPC-DS to AliORC, Apache ORC, or Apache Parquet. Unit: seconds.
- Reader elapsed time: the time required for AliORC, Apache ORC, or Apache Parquet to scan all data in the dataset. Unit: seconds.
- File size: the volume of data stored. The value is the size of all tables. Unit: bytes.
- The following table describes the comparison results based on the store_sales table, which is the largest table among the test tables.

Storage format	Duration of a data write (writer elapsed time)	Duration of a data read (reader elapsed time)	Volume of data stored (file size)
AliORC and Apache ORC	AliORC reduces the duration by more than 86%.	AliORC reduces the duration by more than 74%.	AliORC reduces the data volume by more than 7%.
AliORC and Apache Parquet	AliORC reduces the duration by more than 54%.	AliORC reduces the duration by more than 30%.	AliORC reduces the data volume by more than 20%.

The following figure shows the test results of the store_sales table.



- The following figure compares the test results of compression ratios for 1 TB of data and 10 TB of data based on TPC-DS tests.

table name	10TB tpch rows (official)	10TB odps Table size	1TB tpch rows (official)	1TB odps Table size
call_center	54	9700	42	8205
catalog_page	40000	1273599	30000	958259
catalog_returns	1440033112	83023837172	143996756	8121389060
catalog_sales	14399964710	738936121856	1439980416	73238043325
customer	65000000	2573064032	12000000	471211762
customer_address	32500000	347970480	6000000	64111742
customer_demographics	1920800	68663	1920800	68663
date_dim	73049	156661	73049	156661
household_demographics	7200	1112	7200	1112
income_band	20	605	20	605
inventory	1311525000	4413077985	783000000	2541325306
item	402000	23630410	300000	17644914
promotion	2000	56363	1500	43379
reason	70	1014	65	1002
ship_mode	20	1373	20	1373
store	1500	86712	1002	58725
store_returns	2879970104	126581984476	287999764	12437664912
store_sales	28799983563	965289815084	2879987999	95931184314
time_dim	86400	62444	86400	62444
warehouse	25	3070	20	2833
web_page	4002	43037	3000	32452
web_returns	720020485	40810104745	71997522	3987025583
web_sales	7199963324	372506890826	720000376	36899421656
web_site	78	10995	54	8565
total		2334508272414		233710426852
		2.3TB		233G
Compression ratio		4.34		4.34

Test environments

- Apache Parquet version: Apache Arrow C++ V0.16.0
- Apache ORC version: C++ V1.6.2
- Dataset: TPC-DS 10 GB (SF=10)

Dataset

TPC-DS is a decision support benchmark that uses multi-dimensional data models, such as star and snowflake data models. The benchmark contains 7 fact tables and 17 dimension tables, with an average of 18 columns per table. The tables contain skewed data and values to simulate a real scenario. TPC-DS provides the best test set to measure the performance of SQL-on-Hadoop and the different versions of Hadoop.

The following 24 tables of the TPC-DS dataset are used in this test:

```
store_sales
catalog_sales
inventory
web_sales
store_returns
catalog_returns
web_returns
customer_demographics
customer
item
customer_address
date_dim
time_dim
catalog_page
household_demographics
promotion
store
web_page
web_site
call_center
reason
warehouse
ship_mode
income_band
```

25.DataWorks

25.1. What is DataWorks?

DataWorks is an end-to-end big data platform based on compute engines such as MaxCompute and E-MapReduce. It integrates all processes from data collection to data display and from data analysis to application running. DataWorks provides various features to help you complete the entire research and development (R&D) process in a quick and effective manner. The entire R&D process involves data integration, data development, data governance, data service provisioning, data quality control, and data security assurance.

DataWorks is an all-in-one solution for collecting, presenting, and analyzing data, and driving application development. It not only supports offline processing, analysis, and mining of large amounts of data, but also integrates core data-related technologies such as data development, data integration, production and operations and maintenance (O&M), real-time analysis, asset management, data quality control, data security assurance, and data sharing. In addition, it provides the DataService Studio and Machine Learning Platform for Artificial Intelligence (PAI) services.

In 2018, Forrester, a globally recognized market research company, named Alibaba Cloud DataWorks and MaxCompute as a world-leading cloud-based data warehouse solution. This solution is by far the only solution from a Chinese company to receive such an acknowledgment. Building on the success of the previous version, DataWorks V2.0 incorporates several new additions, such as workflows and script templates. DataWorks V2.0 supports dual workspaces for development, isolates the development environment from the production environment, adopts standard development processes, and uses a specific mechanism to reduce errors in code.

25.2. Benefits

This topic describes the benefits of DataWorks.

- Powerful computing capabilities

DataWorks integrates with compute engines that can process large amounts of data.

- DataWorks supports join operations for trillions of data records, millions of concurrent jobs, and petabytes (PB) of I/O throughput per day.
- The offline scheduling system can run millions of concurrent jobs. You can configure rules and alerts to monitor the running statuses of nodes in real time.
- DataWorks provides efficient and easy-to-use SQL and MapReduce engines, and supports most standard SQL syntax.
- MaxCompute protects user data from loss, breach, or theft by using multi-layer data storage and access security mechanisms, including triplicate backups, read/write request authentication, application sandboxes, and system sandboxes.

- End-to-end platform

DataWorks provides the graphical user interface (GUI) and allows multiple users to collaborate on a workspace.

- DataWorks integrates all processes from data integration, processing, management, and monitoring to output.
- You can create and edit workflows in a visual manner by using the workflow designer.

- DataWorks provides a collaborative development environment. You can create and assign roles for varying nodes, such as development, online scheduling, maintenance, and data permission management, without locally processing data and nodes.
- Integration of heterogeneous data stores

DataWorks supports batch synchronization of data among heterogeneous data stores at custom intervals in minutes, days, hours, weeks, or months. More than 400 pairs of heterogeneous data stores are supported.
- Web-based software

DataWorks is an out-of-the-box service. You can use it on the Internet or an internal network without the need for installation and deployment.
- Multitenancy

Data is isolated among different tenants. Each tenant controls permissions, processes data, allocates resources, and manages members in a unified and independent manner.
- Intelligent monitoring and alerting

By setting monitoring thresholds, you can control the entire process of all nodes as well as monitor the running status of each node.
- Easy-to-use SQL editor

The SQL editor supports automatic code and metadata completion, code formatting and folding, and pre-compilation. It offers two editor themes. These features ensure a good user experience.
- Comprehensive data quality monitoring

DataWorks allows you to control the quality of data in heterogeneous data stores, offline data, and real-time data. You can check data quality, configure alert notifications, and manage connections.
- Convenient API development and management

The DataService Studio service of DataWorks interacts with API Gateway. This makes it easy for you to develop and publish APIs for data sharing.
- Secure data sharing

DataWorks enables you to de-identify sensitive data before you share it with other tenants, which ensures the security of your big data assets and maximizes their value.

25.3. Architecture

DataWorks is an end-to-end big data platform launched by Alibaba Group, which supports big data processing, management, analysis, mining, sharing, and transmission. It releases you from cluster deployment and management. DataWorks adopts MaxCompute (formerly known as ODPS) as the compute engine to process large volumes of data.

DataWorks is developed based on MaxCompute. DataWorks provides a management console and supports functions such as data processing, management, analysis, and mining.

25.3.1. Service architecture

This topic describes the service architecture of DataWorks.

DataWorks provides the following services:

- Data Integration: supports integration of large amounts of data from heterogeneous data stores to a big data platform.
- DataStudio: supports data warehouse design and whole extract, transform, load (ETL) procedure design.
- Operation Center: supports management and monitoring of online ETL nodes, and supports monitoring of large amounts of nodes and instances based on business baselines.
- DataAnalysis: supports ad hoc queries and data analysis.
- Data Asset Management: supports features such as metadata management and provides data maps, data lineages, and data asset dashboards.
- Data Quality: supports data quality check, monitoring, verification, and grading.
- Data Protection: supports permission management, data management based on security levels, data de-identification, and data auditing.
- DataService Studio: supports data sharing and transmission by using APIs.

25.3.2. System architecture

This topic describes the system architecture of DataWorks.

DataWorks is an end-to-end big data platform that enables you to process data by using services such as Data Integration, DataStudio, Data Asset Management, and DataService Studio. It serves as a basis for upper-layer applications, which satisfies all user requirements.

25.3.3. Security architecture

This topic describes the security architecture of DataWorks.

The security architecture of DataWorks features error proofing, basic security, and optional security tools.

- Error proofing ensures proper running of DataWorks during coding, deployment, and configuration.
- Basic security ensures the security of data for DataWorks by using features such as resource isolation among tenants, user identity verification, authentication, and log auditing.
- Optional security tools in DataWorks allow you to customize security policies for the protection and management of your system and data.

25.3.4. Multitenancy

DataWorks adopts multitenancy.

- Storage and computing resources are scalable. Tenants can apply for resource quotas as needed.
- Tenants are isolated and can manage only their own data, permissions, accounts, and roles. This ensures data security.

25.4. Services

25.4.1. Data Integration

This topic provides an overview of the Data Integration service of DataWorks and describes the features provided by Data Integration.

Overview

Data Integration is a stable and efficient data synchronization service provided by Alibaba Cloud. You can use Data Integration to add data sources to and remove data sources from DataWorks. Data Integration is designed to efficiently transmit and synchronize data between heterogeneous data sources in complex network environments. The following synchronization methods are supported: batch synchronization, full synchronization, and incremental synchronization. Data can be synchronized at an interval of minutes, hours, days, weeks, or months.

Data Integration can read and monitor the data of your database. This service also provides an overview of all data sources, such as relational databases, NoSQL databases, big data databases, and FTP servers.

Supported data sources

Data Integration supports the following data sources:

- Metadata

Data Integration can collect metadata from more than 20 types of common data sources, such as MySQL databases, SQL Server databases, Oracle databases, and MaxCompute projects. Data Integration provides a comprehensive overview of all data assets from the collected metadata to help inventory data assets and synchronize core data.

- Relational databases

Data Integration allows you to read data from and write data to relational databases, such as MySQL, SQL Server, Oracle, PolarDB-X, PostgreSQL, Db2, and ApsaraDB RDS for PPAS.

- NoSQL databases

Data Integration allows you to read data from and write data to NoSQL databases, such as HBase, MongoDB, and Tablestore.

- Massively parallel processing (MPP) databases

Data Integration allows you to read data from and write data to MPP databases, such as HybridDB for MySQL and HybridDB for PostgreSQL.

- Big data databases

Data Integration allows you to read data from and write data to MaxCompute projects and Hadoop Distributed File System (HDFS). Data Integration also allows you to write data to AnalyticDB databases.

- Unstructured data sources

Data Integration allows you to read data from and write data to Object Storage Service (OSS) objects and FTP servers.

 **Note** Data Integration supports data exchanges between more than 400 pairs of data sources.

Real-time synchronization

Data Integration allows you to synchronize data from a data source, such as MySQL, Oracle, Db2, SQL Server, OceanBase, DataHub, LogHub, or Kafka, to MaxCompute, Hologres, Kafka, or DataHub in real time.

Inbound data control

Data Integration supports conversion between different data types and accurately identifies, filters, collects, and displays dirty data to facilitate inbound data control. In addition, Data Integration provides statistics such as the data volume, data throughput, and job duration and detects dirty data in each job.

Fast transmission

Data Integration utilizes the network interface controller (NIC) on each server and adopts a distributed architecture to transmit gigabytes or terabytes of data within a short period of time.

Accurate throttling

Data Integration implements accurate throttling on channels, record streams, and byte streams. Data Integration also supports fault tolerance and allows you to rerun specific or all threads, processes, and jobs.

Synchronization agents

Data Integration provides synchronization agents, which can be used to connect to the servers of data sources and collect data.

Cross-network transmission

Data Integration supports data transmission in complex network environments. For example, Data Integration can transmit data across data centers or virtual private clouds (VPCs).

 **Note** Data Integration uses specific protocols to accelerate the transmission of large amounts of data over a long distance. This way, the stability and efficiency of the data transmission are ensured.

25.4.2. DataStudio

25.4.2.1. Overview

DataStudio is an integrated development environment (IDE) that allows you to develop ETL and data mining algorithms, and build data warehouses in DataWorks.

Before using DataStudio, you need to add data stores by using Data Integration. Then, you can use DataStudio to process the data retrieved from the data stores.

25.4.2.2. Workflows

This topic describes workflows in DataStudio.

Overview

In DataStudio, you can organize various data development nodes in a workflow. DataStudio provides you with a directed acyclic graph (DAG) for nodes in each workflow. It also provides professional tools and supports administrative operations for workflows, which promotes intelligent development and management.

A workflow can contain the following types of nodes: ODPS SQL, ODPS MR, shell, machine learning, data synchronization, PyODPS, SQL component, and zero-load node. You can configure dependencies between nodes within the same workflow or across workflows. You can also schedule a whole workflow or specific nodes.

Manage nodes

You can organize the following types of nodes in a workflow: ODPS SQL, ODPS MR, shell, machine learning, data synchronization, PyODPS, SQL component, and zero-load node. Nodes can be scheduled based on node dependencies or schedules. Each node can depend on other nodes in the current workflow or nodes from other workflows.

Configure a node

After you double-click a node in the left-side navigation pane or in a DAG, the configuration tab of the node appears. Then, you can configure the node. For example, you can write SQL statements for an ODPS SQL node or configure data synchronization rules for a batch sync node. You can also click the tabs in the right-side navigation pane to view the version information or modify settings such as the scheduling properties and lineage of the node.

View the versions of a node

You can view the versions of a node, for example, an ODPS SQL node, an ODPS MR node, or a shell node. If required, you can roll back a node to an earlier version.

Deploy a node

In workspaces in the standard mode, you can deploy nodes that have passed tests to the production environment.

25.4.2.3. Solutions

In a DataWorks workspace, you can group multiple workflows in a solution.

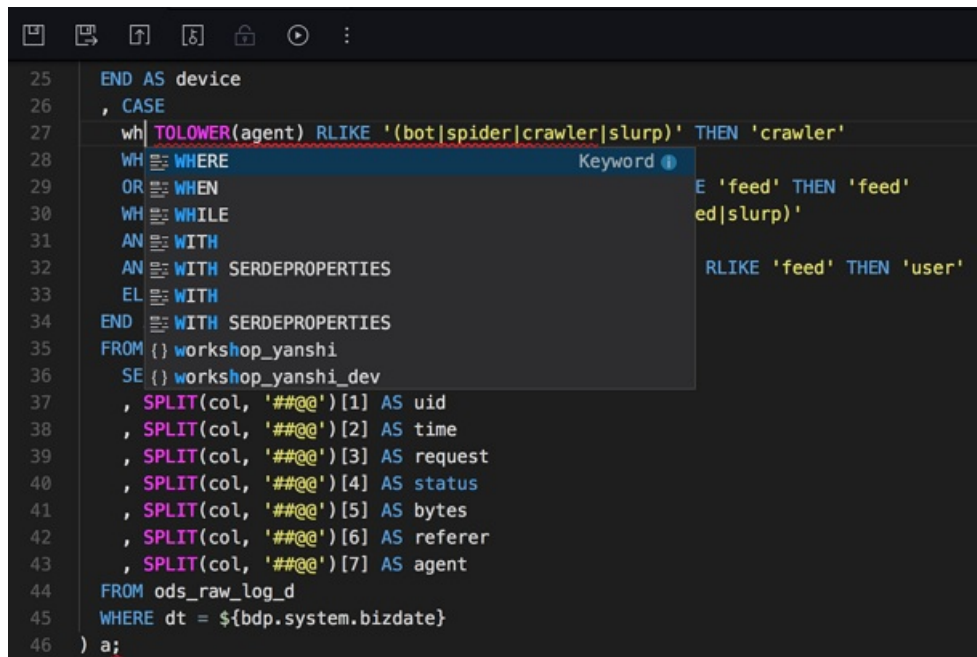
You can add one or more workflows to one solution so that you can manage them as a whole. In addition, a workflow can be added to multiple solutions, allowing you to assess your business based on solutions.

25.4.2.4. Code editor

DataStudio provides a code editor. You can configure ODPS SQL and ODPS MR nodes, upload files as resources, register user-defined functions (UDFs), and write shell scripts in the code editor.

Configure ODPS SQL nodes

The web-based code editor allows you to write SQL statements. It supports a variety of features such as automatic SQL statement completion, code formatting and highlighting, and debugging.



```

25  END AS device
26  , CASE
27  wh| TOLOWER(agent) RLIKE '(bot|spider|crawler|slurp)' THEN 'crawler'
28  WH ≡ WHERE Keyword ⓘ
29  OR ≡ WHEN
30  WH ≡ WHILE
31  AN ≡ WITH
32  AN ≡ WITH SERDEPROPERTIES
33  EL ≡ WITH
34  END ≡ WITH SERDEPROPERTIES
35  FROM {} workshop_yanshi
36  SE {} workshop_yanshi_dev
37  , SPLIT(col, '###')[1] AS uid
38  , SPLIT(col, '###')[2] AS time
39  , SPLIT(col, '###')[3] AS request
40  , SPLIT(col, '###')[4] AS status
41  , SPLIT(col, '###')[5] AS bytes
42  , SPLIT(col, '###')[6] AS referer
43  , SPLIT(col, '###')[7] AS agent
44  FROM ods_raw_log_d
45  WHERE dt = ${bdp.system.bizdate}
46  ) a;
  
```

Configure ODPS MR nodes

When you configure an ODPS MR node in the code editor, you can upload a Java Archive (JAR) file that contains MapReduce code as a JAR resource and then reference the file in the node.

Upload files as resources

DataWorks supports the following types of resources:

- JAR: You can upload JAR files as file resources. Then, UDFs or ODPS MR nodes can reference the resources.
- Python: You can upload Python files as Python resources. Then, UDFs can reference the resources.
- File: You can upload user-defined files such as shell scripts, XML configuration files, or TXT configuration files as file resources.
- Archive: You can upload compressed files as archive resources. The following file formats are supported: .zip, .tgz, .tar.gz, .tar, and .jar. DataWorks automatically identifies the format of an uploaded file based on the file name extension.

Register UDFs

You can register Java or Python UDFs in the code editor. Before you register UDFs, you must upload JAR or Python files as resources. Then, the UDFs can reference the resources.

Write shell scripts

You can use the code editor to write and debug shell scripts online.

25.4.2.5. Code repository and team collaboration

DataWorks allows multiple users to simultaneously work on the same workspace, which improves development efficiency.

DataWorks adopts a lock mechanism that allows you to lock workflows and nodes. This ensures that each workflow or node is edited by only one user at the same time. To edit a node that is locked by another user, you can force unlock the node and then lock the node yourself. This operation is called steal lock. After you steal the lock of a node, the system sends a notification to the user who locked the node previously.

In addition, DataWorks records each committed version of your node and workflow. You can compare two versions of a node and roll back a node to an earlier version.

25.4.3. Administration

25.4.3.1. Overview

Operation Center is a centralized data operations and management platform for data developers and administration experts. You can control and monitor the running of nodes and instances, and set node priorities in Operation Center.

Due to the volume, diversity, and complexity of data used in DataWorks, it is necessary to use a scheduling system that supports high concurrency, multiple cycles, and various data processing procedures.

Operation Center allows you to trace all the nodes that are committed to the scheduling system, view alerts when nodes do not run as scheduled or fail, and view daily reports of node statistics.

25.4.3.2. Overview page

The Overview page displays running statistics of nodes and instances.

You can view the following statistics on the Overview tab: the trend of node instances run today and in past days, rankings of nodes sorted by duration, by number of errors, and by number of overtime node instances within 30 days, and the distribution of nodes by status and by type.

25.4.3.3. Node O&M pages

You can view a node in a directed acyclic graph (DAG), which allows you to perform operations and maintenance (O&M) in a visual manner.

- You can rerun, stop, or suspend nodes, set the status of nodes to successful, and configure alerts to monitor the running status of nodes.
- You can view each node in a node list or the DAG. The DAG clearly shows the relationships between nodes.
- You can view the running status of auto triggered nodes, test nodes, and manually triggered nodes.
- You can view the operational logs, code, and property settings of nodes.

25.4.3.4. Intelligent Monitor service

Intelligent Monitor is a system that monitors and analyzes nodes in DataWorks.

Intelligent Monitor monitors the running status of nodes and sends alerts based on the intervals, notification methods, and recipients specified in alert triggers. When the alerting condition is met, Intelligent Monitor automatically selects the most appropriate alerting time, notification methods, and recipients.

Intelligent Monitor provides you with the following benefits:

- Improves your efficiency on configuring monitoring rules.
- Prevents invalid alerts.
- Automatically covers all important nodes.

Intelligent Monitor provides comprehensive monitoring and alerting logic. You only need to provide the names of important nodes in your business. Then, Intelligent Monitor automatically monitors the entire process of your nodes and creates standard alert triggers for them. Intelligent Monitor also allows you to customize the monitoring feature. You can define alert triggers based on your business requirements.

25.4.3.5. Engine O&M

This topic provides an overview of the engine O&M feature provided by DataWorks.

DataWorks provides some views that allow you to manage the resources of compute engines in end-to-end mode. For example, you can view the usage details of computing and storage resources and the usage details of the resources that are occupied by jobs. You can use the engine O&M feature to view the details about a job and identify and remove the jobs that fail to run. This prevents failed jobs from affecting the running of the DataWorks node instances to which the jobs belong and the descendant nodes of the nodes that generate the node instances.

25.4.4. DataAnalysis

DataAnalysis provides two core features: ad hoc query and private table management. It expedites the analysis process by using the data collection tools of MaxCompute in the near real-time mode.

Benefits

By default, the near real-time mode is used.

You can run the `set ODPS.service.mode=[all|off|limited]` command to change the configuration.

The near real-time mode has the following advantages over the standard mode:

- In the near real-time mode, DataAnalysis preallocates thread pools based on the job size. The near real-time mode eliminates the need for Job Scheduler to plan jobs and reduces the preparation time to run jobs.
- In the near real-time mode, DataAnalysis shuffles data from Mappers to Reducers, without transmitting the data to Apsara Distributed File System.

Keynotes

- The near real-time mode is used if you set the ODPS.service.mode parameter to all. However, if MaxCompute resources are insufficient to run SQL nodes, DataAnalysis switches to the standard mode in which Job Scheduler is responsible for resource allocation. For example, Time Analysis switches to the standard mode if insufficient workers are available for creating instances.
- The scheduling process in the near real-time mode is still complex, but is much more time-saving than the scheduling process in the standard mode.
- If you set the ODPS.service.mode parameter to all, DataAnalysis preferentially uses the near real-time mode. DataAnalysis uses the standard mode if system resources are insufficient, or if known issues or unknown exceptions occur in the near real-time mode.

25.4.5. Data Map

This topic describes the features provided by the Data Map service.

You can use Data Map to manage the metadata and data assets of your business. You can also use Data Map to globally search for data, view the details about the metadata, preview data, view data lineage, and manage data categories. Data Map can help you search for, understand, and use data.

25.4.6. Security Center

Security Center allows you to manage permissions with ease and submit and handle requests on a visual interface.

Security Center provides the following features on the **My Permissions**, **Authorizations**, and **Approval Center** pages:

- Self-service permission requesting: Users can select the tables on which the users want to request permissions. Then, the users can submit permission requests online. This online request mode is more efficient than the original mode in which users need to contact administrators offline.
- Permission management: Administrators can view the users who have permissions on database tables. If the permissions of the users do not meet specific requirements, administrators can revoke the permissions. Users can also manually remove the permissions that the users no longer require.
- Permission request handling: Administrators cannot directly grant permissions to users. Instead, administrators must handle the permission requests submitted by users. A visual and process-oriented mechanism to manage permissions is implemented. This mechanism allows users to review handling processes.

In Security Center, you can view the permissions on all the tables that belong to a tenant, request and manage permissions on tables, and approve or reject permission requests.

25.4.7. DataService Studio

DataService Studio aims to build a data service bus to help enterprises manage private and public APIs in a unified manner.

DataService Studio allows you to create APIs based on data tables. You can also register existing APIs to DataService Studio for unified management. DataService Studio and API Gateway are interconnected. This allows you to publish APIs to API Gateway with ease. DataService Studio, together with API Gateway, provides a secure, stable, cost-effective, and easy-to-use API development and management service. DataService Studio adopts a serverless architecture. This allows you to focus on the query logic of the API without worrying about the infrastructure, such as compute resources. DataService Studio supports automatic scaling for compute resources, which significantly reduces your operations and maintenance (O&M) costs.

DataService Studio serves the government as a secure, flexible, and reliable platform for data sharing across departments and networks within the government. It also enables the government to share data with the public.

Create an API operation

DataService Studio allows you to create APIs based on tables in relational databases, NoSQL databases such as Tablestore, and analytical databases such as AnalyticDB. You can create an API in the codeless UI within a few minutes without the need to write code. You can call an API immediately after it is created. DataService Studio also allows you to create an API in the code editor. You can write SQL statements to customize the query logic of the API. In the code editor, you can specify multi-table join queries, complex query criteria, and aggregate functions.

Register an API

You can register existing RESTful APIs to DataService Studio to manage them together with the APIs that are created in DataService Studio based on tables. Four request methods and three data formats are supported. The four request methods are GET, POST, PUT, and DELETE. The three data formats are tables, JSON, and XML.

API Gateway

API Gateway provides API lifecycle management services, including API publishing, management, maintenance, and monetization. API Gateway helps you integrate microservices, separate the frontend from the backend, and integrate systems at low costs and low risks in an easy and quick manner. API Gateway enables you to share features and data with partners and developers. Being integrated with API Gateway, DataService Studio allows you to publish APIs to API Gateway conveniently. Both APIs that you create based on data tables and APIs that you register to DataService Studio can be published to API Gateway for management, for example, for authorization, authentication, throttling, and billing.

25.4.8. Migration Assistant

The Migration Assistant service of DataWorks allows you to migrate data objects across different DataWorks versions, Alibaba Cloud accounts, regions, and workspaces.

Migration Assistant allows you to perform cross-cluster migration, cross-version migration, or custom migration on your data objects. The data objects include auto triggered nodes, manually triggered nodes, resources, functions, data sources, script templates, ad hoc queries, table metadata, and objects in DataService Studio.

Scenarios

- Back up node code

You can use Migration Assistant to back up your node code on a regular basis. This prevents data from being deleted by mistake.

- Export common workflows for replication

You can use Migration Assistant to export common workflows that can be replicated.

- Build a test environment

You can use Migration Assistant to replicate all node code and replace production data with test data to build a test environment.

- Process data in a hybrid cloud environment

You can use Migration Assistant to migrate node code from Alibaba Cloud public cloud to Alibaba Cloud Apsara Stack to process data in a hybrid cloud environment.

- Migrate data objects between development and production environments

If the production environment of your workspace is completely isolated from the development environment of the workspace, you can use Migration Assistant to export nodes from the development environment and import the nodes to the production environment.

25.4.9. Workspace Management

Workspace Management enables administrators to manage their organizations and workspaces.

Workspaces are organizational units for code, member, role, and permission management in DataWorks. Workspaces are isolated from each other. You can view and modify code in a workspace only if you are a member of the workspace and have been granted the required permissions.

 **Note** A user can be a member of multiple workspaces at the same time. The user's permissions in each workspace vary based on the role assigned to the user.

Workspace Management provides the Organizations, Workspaces, Members, and Authorizations pages for managing organizations, workspaces, members, and permissions, respectively.

Organizations

The Organizations page displays the account, AccessKey ID, and AccessKey secret of the owner of the current organization. On this page, you can manage all members in the organization.

Workspaces

On the Workspaces page, you can create, modify, activate, and disable workspaces.

Members

The **Members** page displays information, such as the name, logon username, and roles, about each member of the current workspace. On this page, you can perform the following operations:

- Search for workspace members in the fuzzy match mode and remove the target members from the current workspace.
- Search for users in the fuzzy match mode and add the target users as members of the current workspace.

 **Note** When you add a user as a member of a workspace, you must assign at least one role to the user.

Only workspace administrators can add members to and remove members from workspaces.

 **Note** After a user is removed from a workspace, all permissions that have been granted to the user within the workspace are revoked.

Authorizations

On the Authorizations page, you can manage roles and specific permissions for all users.

The following table describes the permissions of each role in DataWorks.

Role	Permissions
Administrator	An administrator can manage the basic properties, data stores, compute engine configurations, and members of the workspace. The administrator can also assign the administrator, developer, administration expert, deployment expert, and visitor roles to other members of the workspace.
Developer	A developer can create workflows, script files, resources, and user-defined functions (UDFs). The developer can also create and delete tables, and create deployment tasks. However, the developer cannot perform deploy operations.
Administration expert	An administration expert can perform deploy and administrative operations, but does not have the permissions of a developer. The administration permissions of an administration expert are assigned by an administrator.
Deployment expert	A deployment expert can perform all operations that an administration expert can, except administrative operations.
Visitor	A visitor can only view data, but cannot edit workflows or code in workspaces.

25.4.10. Data Asset Management

Data Asset Management is a tenant-level feature. To use this feature, you must first obtain required permissions on the Project Management page.

This feature allows you to manage your data assets, such as tables and APIs, in your business system and DataWorks. Before you use this feature, you must use Data Integration to synchronize data and then use DataStudio to process the data.

25.4.11. Data Protection

25.4.11.1. Overview

Data Protection is a data security management platform. It can be used to identify data assets, detect sensitive data, classify data, de-identify data, monitor data access behavior, report alerts, and audit risks.

Data Protection provides security management services for MaxCompute.

Data Protection provides the following features:

- Sensitive data detection

Data Protection automatically detects an enterprise's sensitive data based on self-training models and algorithms, and displays statistics on data types, volume, and visitors. It also recognizes custom data types.

- Custom data classification

Data Protection allows you to classify data and create custom levels for better data management.

- Flexible data de-identification

Data Protection provides diverse and configurable methods for dynamic data de-identification.

- Monitoring and auditing of risky user behavior

Data Protection uses various association analysis algorithms to detect risky user behavior. It also provides alerts and supports visualized auditing for detected risks.

25.4.11.2. Terms

This topic describes the terms that are used in Data Protection, for example, organization, workspace, and data de-identification.

Organization

An organization refers to all system settings and resources owned by a single tenant in DataWorks. The system settings and resources include account configurations, permission settings, and custom applications.

Workspace

Workspaces are organizational units in DataWorks. Similar to databases in a relational database management system (RDBMS), workspaces isolate resources among different users and offer boundaries for access control. Tables, resources, user-defined functions (UDFs), and nodes are isolated among different workspaces.

Regular expression

A regular expression is a sequence of characters that define a search pattern. You can use regular expressions to detect sensitive data.

 **Note** A regular expression consists of metacharacters and literal characters such as letters from a to z.

Data classification

Data is classified based on value, sensitivity, related risks, legal and regulatory requirements, and the potential impact of data breaches.

Sensitive data detection

Data Protection detects sensitive data on the user side based on user-defined rules.

Data de-identification

Data Protection de-identifies sensitive data based on user-defined rules.

MaxCompute

MaxCompute is a data processing platform developed by Alibaba Cloud for large-scale data warehousing. Being able to store and compute mass structured data, MaxCompute provides support for various data warehouse solutions as well as big data analysis and modeling.

25.4.11.3. Management

You can configure sensitive data detection rules on the Data Definition page as a security expert.

After you configure sensitive data detection rules, you can go to the **Data Recognition Rules** page or the **Manipulations and Queries** or **Export** tab of the Data Activities page to perform relevant operations.

25.4.11.4. Data recognition

On the next day after you configure data recognition rules, you can view the recognized data in the Overview, Level, and Fields Recognized sections on the Data Recognition page.

You can filter recognized data by project, rule name, rule type, and risk level.

25.4.11.5. Data Activities

This topic describes data activities.

Data activities include data manipulations and queries, and data export.

- Data manipulations and queries include successful create, insert, and select operations that are performed on data.
- Data export refers to the operation of exporting data from MaxCompute.

Manipulations and queries

On the next day after you configure sensitive data detection rules as a security expert, you can view data manipulations and queries on the Manipulations and Queries tab of the Data Activities page. The Manipulations and Queries tab displays information about data access activities, including the overview, trend, and records. You can filter the information by project, user, rule name, rule type, and risk level based on your query requirements.

Export

On the next day after you configure data recognition rules as a security expert, you can view data export activities on the Export tab of the Data Activities page. The Export tab displays information about data export from MaxCompute, including the overview, top N accounts that have exported the most data, and data export details. You can filter the information by rule name, rule type, and greater than condition based on your query requirements.

25.4.11.6. Data masking

On the Data Masking page, you can create, modify, delete, and test data masking rules.

You can configure data masking rules for each data recognition rule, and configure a whitelist to include recognized sensitive data that does not require data masking.

25.4.11.7. Levels

On the Levels page, you can configure the security levels of rules if the existing configuration cannot meet your needs.

You can create levels, delete levels, and adjust the priority of levels and rules on the **Levels** page.

25.4.11.8. Manual check

On the Manual Check page, you can manually modify recognition results if any sensitive data is recognized incorrectly. You can delete data that is incorrectly recognized, change the type of recognized data, and process multiple data records at a time.

25.4.11.9. Data risks

In Data Protection, data activities are audited manually or based on the risk identification rules and AI-based identification rules configured on the Risk Rules page. The Data Risks page lists data activities that are audited as risky. You can also comment audit results as required.

25.4.11.10. Risk Rules

The Risk Rules page allows you to configure risk identification rules.

You can configure risk identification rules or enable AI-based identification rules to identify risks in users' daily access to your data. The Data Risks page lists the data activities where risks are identified. You can check these data activities and mark them as secure or risky. On the Data Activities page, you can click an activity to view the risk rule that the activity hits.

25.4.11.11. Data Auditing

The Data Auditing page provides an overview and the trend of the total number data risks, number of data risks that have been handled, and number of data risks that have not been handled. This page also provides risk analysis from multiple dimensions.

You can view the data in the **Total Risks**, **Risks Handled**, **Risks Not Handled**, **Trend**, and **Risk Analysis by Dimension** sections.

25.5. Scenarios

25.5.1. Cloud-based data warehouse

Enterprises can use DataWorks in Apsara Stack to build large data warehouses.

DataWorks provides superior data processing capabilities:

- Mass storage: supports petabyte- and exabyte-level data warehouses and scalable storage.
- Data integration: supports data synchronization and integration across heterogeneous data stores to eliminate data siloes.
- Data analytics: supports MaxCompute-based big data analytics, programming frameworks such as SQL and MapReduce, and a visualized workflow designer.
- Data management: supports unified metadata management and permission-based data access control.
- Batch scheduling: supports real-time node monitoring and error alerts, periodic node execution, and processing for millions of nodes per day.

25.5.2. Business intelligence

This topic describes how to create reports by using DataWorks.

You can analyze the following items based on the network logs of your website:

- Page views, unique visitors, and device types such as Android devices, iPads, iPhones, and PCs. You can also create a daily report based on these statistics.
- Locations of visitors.

The following log entry is used as an example:

```
xx.xxx.xx.xxx - - [12/Feb/2014:03:15:52 +0800] "GET /articles/4914.html HTTP/1.1" 200 37666  
"http://xxx.cn/articles/6043.html" "Mozilla/5.0 (Windows NT 6.2; WOW64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/xx.x.xxxx.xxx Safari/537.36" -
```

1. Create a destination table named ods_log_tracker in MaxCompute, and then import data to the table.
2. Configure dependencies among the tables to be analyzed.
3. Create a shell node, a sync node, and an ODPS SQL node.
4. Configure the created nodes.

25.5.3. Data-driven management

- Innovative business: Data mining, data modeling, and real-time decision making can be implemented based on big data analytics results provided by DataWorks.
- Small and medium enterprises: With DataWorks, data can be quickly analyzed and put into commercial use, which helps enterprises to generate operational strategies.

25.6. Limits

None.

25.7. Terms

This topic introduces the terms related to DataWorks, including workspace, workflow, solution, SQL script template, node, instance, commit operation, script, resource, function, and output name.

workspace

Workspaces are basic units for managing nodes, members, roles, and permissions in DataWorks. The administrator of a workspace can add members to the workspace and assign the workspace administrator, developer, administration expert, deployment expert, security expert, or visitor role to each member. This way, workspace members with different roles can collaborate with each other.

 **Note** We recommend that you create workspaces by department or business unit to isolate resources.

You can bind multiple types of compute engine instances to a workspace, such as MaxCompute, E-MapReduce (EMR), and Realtime Compute for Apache Flink. After you bind compute engine instances to a workspace, you can configure and schedule nodes in the workspace.

workflow

Workflows are abstracted from business to help you manage and develop code based on business requirements and improve the efficiency of node management.

 **Note** A workflow can be used in multiple solutions.

Workflows help you manage and develop code based on business requirements. A workflow has the following features:

- Allows you to organize nodes by type.
- Supports a hierarchical directory structure. We recommend that you create a maximum of four levels of subdirectories for a workflow.
- Allows you to view and optimize a workflow from the business perspective.
- Allows you to deploy and manage nodes in a workflow as a whole.
- Provides a dashboard for you to develop code with improved efficiency.

solution

A solution contains one or more workflows.

Solutions have the following benefits:

- A solution can contain multiple workflows.
- A workflow can be used in multiple solutions.
- Workspace members can collaboratively develop and manage solutions in a workspace.

SQL script template

SQL script templates are general logic chunks that are abstracted from SQL scripts. They can help reuse code.

Each SQL script template involves one or more source tables. You can filter source table data, join source tables, and aggregate source tables to generate a result table based on your business requirements. An SQL script template contains multiple input and output parameters.

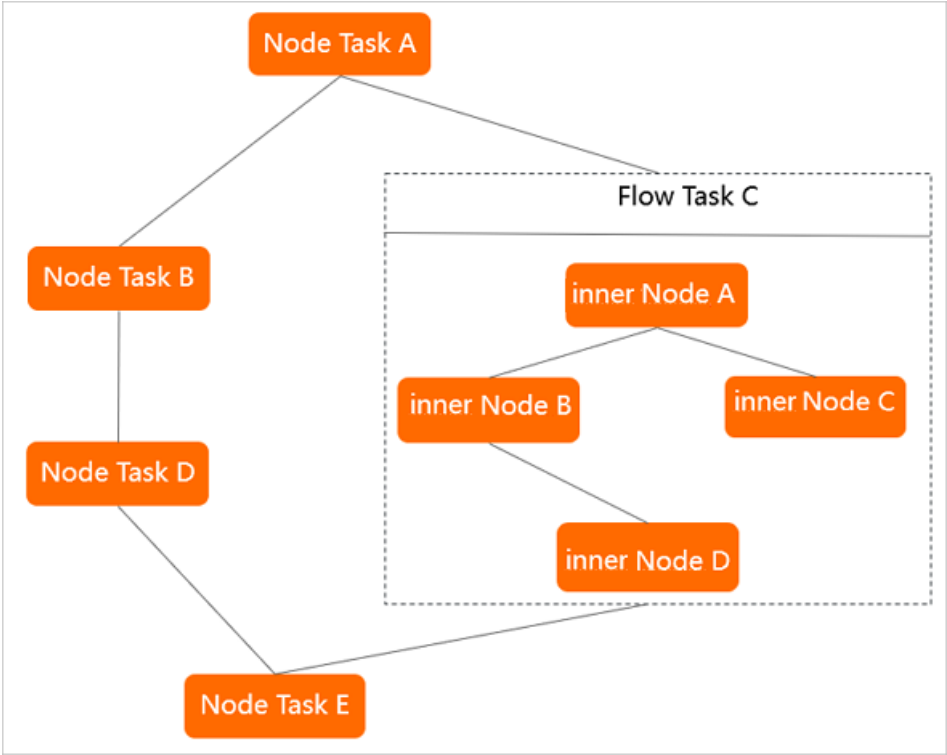
node

Each type of node is used to perform a specific data operation. Examples:

- A synchronization node can be used to synchronize data from ApsaraDB RDS to MaxCompute.
- A MaxCompute SQL node can be used to convert data by executing SQL statements that are supported by MaxCompute.

Each node has zero or more input tables or datasets and generates one or more output tables or datasets.

Nodes are classified into node tasks, flow tasks, and inner nodes.



Node type	Description
Node task	A node task is used to perform a data operation. You can configure dependencies between a node task and flow tasks or other node tasks to form a directed acyclic graph (DAG).
Flow task	A flow task contains a group of inner nodes that process a workflow. We recommend that you create less than 10 flow tasks. Inner nodes in a flow task cannot be the dependencies of node tasks or other flow tasks. You can configure dependencies between a flow task and node tasks or other flow tasks to form a DAG.  Note In DataWorks V2.0 and later, you can find the flow tasks that are created in DataWorks V1.0 but cannot create flow tasks. Instead, you can create workflows to perform similar operations.
Inner node	An inner node is a node within a flow task. The features of an inner node are basically the same as those of a node task. You can drag lines between inner nodes in a flow task to configure dependencies. However, you cannot configure a recurrence for inner nodes because they follow the recurrence configuration of the flow task.

instance

An instance is a snapshot of a node at a specific point in time. An instance is generated each time a node is run as scheduled by the scheduling system or is manually triggered. An instance contains information such as the time at which the node is run, the running status of the node, and operational logs.

For example, Node 1 is configured to run at 02:00 every day. The scheduling system automatically generates an instance for Node 1 at 23:30 every day. At 02:00 the next day, if the scheduling system verifies that all the ancestor instances are run, the system automatically runs the instance of Node 1.

 **Note** You can query the instance information on the **Cycle Instance** page in **Operation Center**.

commit

You can commit nodes and workflows from the development environment to the scheduling system. The scheduling system runs the code in the committed nodes and workflows based on configurations.

 **Note** The scheduling system runs only committed nodes and workflows.

script

A script stores code for data analysis. The code in a script can be used only for data queries and analysis. It cannot be committed to the scheduling system for scheduling.

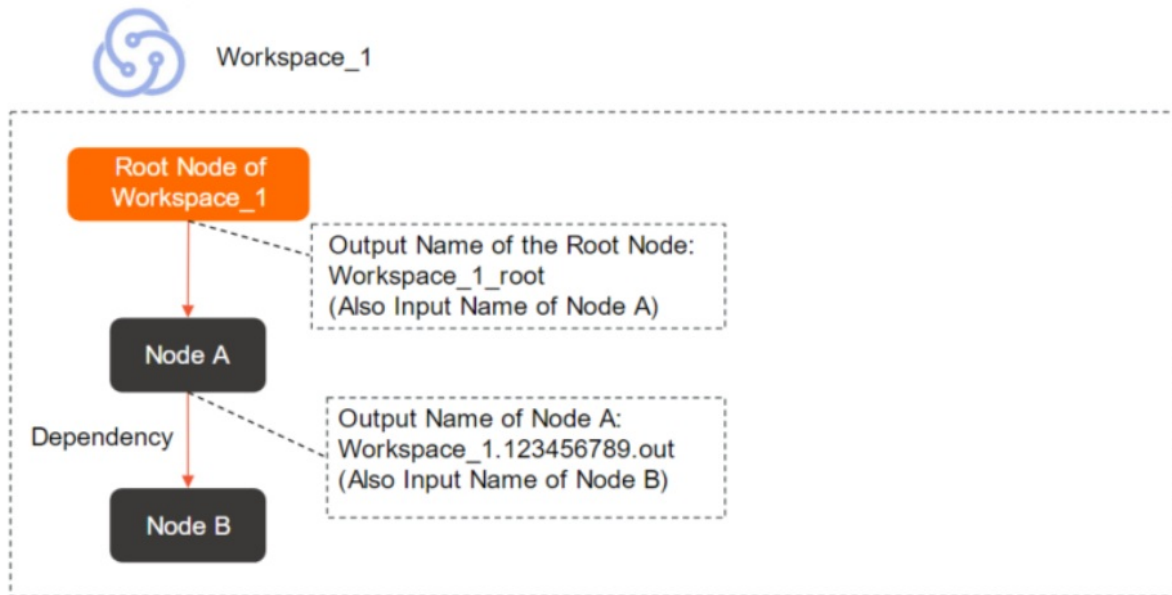
resource and function

Resources and functions are concepts in MaxCompute. You can manage resources and functions in the DataWorks console. If resources or functions are uploaded by using other services such as MaxCompute, you cannot query them in DataWorks.

output name

Within an Apsara Stack tenant account, each node has an output name that is used to connect to its descendant nodes.

When you configure dependencies for a node, you must use its output name instead of its node name or ID. After you configure the dependencies, the output name of the node serves as the input name of its descendant nodes.



Note Each output name distinguishes a node from other nodes under the same Apsara Stack tenant account. By default, an output name is in the following format: Workspace name.Randomly generated nine-digit number_out. You can customize the output name for a node. Note that the output name of each node must be unique within an Apsara Stack tenant account.

26.Realtime Compute

26.1. What is Realtime Compute?

Alibaba Cloud Realtime Compute is an advanced stream processing platform that provides real-time computations over data streams.

Background information

As the demands for high data timeliness and operability increase, software systems need to process more data in less time. In traditional models of big data processing, online transaction processing (OLTP) and offline data analysis are separately performed at different times. These traditional models follow the scheduled processing mode, which accumulates and processes data in a computing cycle that can last hours or even days. Realtime Compute comes from the strict demand for the timeliness of data processing. The business value of data decreases as time passes. Therefore, data must be computed and processed as soon as possible after it is generated. Traditional data processing models cannot satisfy the growing demand for computing data streams. A delay in data processing may cause great impacts in delay-sensitive scenarios such as real-time big data analytics, risk control and alerting, real-time prediction, and financial transactions. To address this issue, Alibaba Cloud provides Realtime Compute to perform real-time computations over data streams.

Realtime Compute shortens the data processing delay, provides a real-time computational logic, and greatly reduces computing costs to help meet the business needs for real-time processing of big data. In addition, Realtime Compute can store audit logs and automatically dump them to a specified server directory for long-term storage and management.

Streaming data

Big data can be viewed as a series of discrete events. These discrete events form event streams or data streams along a timeline. Streaming data has a smaller scale than offline data. Streaming data is generated from continuous event streams, including:

- Log files
- Online shopping data
- In-game player activity information
- Social network information
- Financial transaction information
- Geospatial service information
- Telemetry data from devices or instruments

Features

- Real-time and unbounded data streams

Realtime Compute processes data streams in real time. Streaming data is continuously generated from data sources and is subscribed and consumed in chronological order. For example, when Realtime Compute processes data streams from website visit logs, the log data streams continuously enter the Realtime Compute system if the website is online.

- Continuous and efficient computing

Realtime Compute is an event-driven system where unbounded event or data streams continuously trigger real-time computations. Each streaming data record triggers a computational task. Realtime Compute performs continuous and real-time computations on data streams.

- Real-time integration of streaming data

Realtime Compute writes the computing result of each streaming data record into the destination data store in real time. For example, the system can directly write the computed report data to an ApsaraDB RDS instance to display reports. Realtime Compute continuously writes the result data into the destination data store in real time. Therefore, Realtime Compute can be viewed as a data source that generates data streams for the destination data store.

26.2. End-to-end real-time computing

Unlike offline or batch computing, end-to-end real-time computing of Alibaba Cloud runs real-time computations over data streams, including real-time data collection, computing, and integration. The real-time computational logic of Realtime Compute ensures a short processing delay.

1. Data collection

You can use data collection tools to collect and send streaming data in real time to a publish-subscribe system for big data analysis. This publish-subscribe system continuously produces events for Realtime Compute in the downstream to trigger stream processing jobs.

2. Stream processing

Data streams continuously enter Realtime Compute for real-time computing. At least one data stream must enter the Realtime Compute system to trigger a real-time computing job. Each batch of incoming data records initiates a stream processing procedure in Realtime Compute. The computing results for each batch of data records are then instantly provided.

3. Data integration

Realtime Compute allows you to write the result data of stream processing to sinks, such as tables of data stores and message delivery systems. You can also integrate Realtime Compute with the alerting system that is connected to your business applications. This enables you to easily receive alerts if the specified business rules for alerting are satisfied. Unlike batch computing products such as MaxCompute and open source Apache Hadoop, Realtime Compute inherently comes with data integration modules that allow you to write result data to sinks.

4. Data consumption

After the result data of stream processing is written to sinks, the data consumption phase is decoupled from real-time computing. You can use data stores, data transmission systems, or alerting systems to access the result data, send and receive the result data, or send alerts, respectively.

26.3. Differences between real-time computing and batch computing

26.3.1. Overview

Compared with batch processing, stream processing is an emerging technology in the field of big data computing. This section describes the differences between batch processing and stream processing from two aspects: users and products.

 **Note** For more information, see [Stream processing](#) from Wikipedia.

26.3.2. Batch computing

Batch computing models have been used for most traditional data computing and analysis services. In batch computing models, extract-transform-load (ETL) or online transaction processing (OLTP) systems are used to load data into data stores. The loaded data is then used for online data services, such as ad-hoc queries and dashboard services, based on SQL statements. You can also use SQL statements to obtain results from the analysis.

Batch computing models are widely accepted along with the evolution of relational databases in diversified industries. However, in the era of big data, with the increasing number of human activities being converted to information and then data, more and more data requires real-time and stream processing. The current processing models are facing great challenges in real-time processing.

A typical batch computing model is described as follows:

1. An ETL or OLTP system is used to build data stores and provides raw data for computing and analysis. The batch computing model where users load the data and the batch computing system optimizes queries on the loaded data using multiple methods, such as creating indexes, based on its storage and computing capabilities. In batch computing models, data must be loaded into the batch computing system. Newly arriving data records are collected into a batch and the entire batch is then processed after all data in the batch is loaded.
2. A user or system initiates a computing job, such as a MaxCompute SQL job or Hive SQL job, and submits requests to the ETL or OLTP system. The batch computing system then schedules computing nodes to perform computations on large amounts of data. This may take several minutes or even hours. The mechanism of batch computing determines that the data to be processed is the accumulated historical data. As a result, the data processing may not be in real time. In batch computing, you can change computational logic using SQL at any time to meet your needs. You can also perform ad-hoc queries instantly after changing the logic.
3. The computing results are returned in the form of data sets when a computing job is completed. If the size of result data is excessively large, the result data is stored in the batch computing system. In this scenario, you can integrate the batch computing system with another system to view the result data. Large amounts of result data lead to a lengthy process of data integration. The process may take several minutes or even hours.

Batch computing jobs are initiated by users or systems and are processed with a long delay. The batch computing procedure is described as follows:

1. You load data into the data processing system.
2. You submit computing jobs. In this phase, you can change computing jobs to meet your business needs, and publish the changed jobs.
3. The batch computing system returns the computing results.

26.3.3. Real-time computing

Unlike batch computing, real-time computing runs real-time computations over data streams and allows for a low processing delay. The differences between real-time computing and batch computing are described as follows:

1. Data integration. For real-time computing, data integration tools are used to send streaming data in real time to streaming data stores such as DataHub. For batch computing, large amounts of data

are accumulated and then processed. In contrast, streaming data is sent in micro batches in real time, which ensures a short delay for data integration.

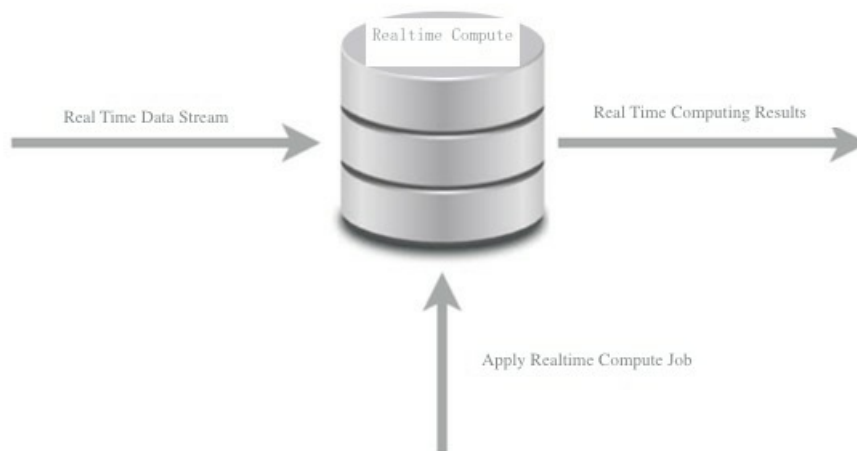
The streaming data is continuously written to data stores in real time. You do not need to preload data for processing. Realtime Compute does not store real-time data that is continuously processed. The real-time data is discarded instantly after it has been processed.

2. Data computing. For batch computing, data is processed only after large amounts of data have been accumulated. In contrast, a real-time computing job is resident in the system and waits to be triggered by events once it is started. Each incoming micro batch of streaming data records initiates a real-time computing job. The computing results are instantly provided by Realtime Compute. Realtime Compute also divides large batches of data records into smaller batches for incremental computing. This effectively shortens the processing delay.

For real-time computing, you must predefine the computational logic in Realtime Compute. You cannot change the computational logic when real-time computing jobs are running. If you terminate a running job and publish the job after changing the computational logic, the streaming data that has been processed before the change cannot be processed again.

3. Writing result data to target systems. For batch computing, result data can be written to online systems by batch only after all accumulated data has been processed. In contrast, real-time computing allows for writing result data to online and offline systems instantly after each micro batch of data records has been processed. This allows you to view the computing results in real time.

Realtime computing



Realtime Compute runs real-time computations over data streams, which are continuously generated from data sources, based on an event-driven mechanism. Realtime Compute allows you to process data streams with a short delay. The real-time computing procedure is described as follows:

1. You publish real-time computing jobs.
2. Streaming data triggers real-time computing jobs.
3. Realtime Compute constantly returns the computing results.

26.3.4. Comparison between real-time computing and batch computing

Comparison between real-time computing and batch computing shows the differences between real-time computing and batch computing.

Comparison between real-time computing and batch computing

Item	Batch computing	Real-time computing
Data integration	You load data into the data processing system.	Data is loaded and processed in real time.
Computational logic	The computational logic can be changed, and data can be reprocessed.	After the computational logic is changed, data cannot be reprocessed. This is because streaming data is processed in real time.
Data scope	You can query and process all or most of the data in the data set.	You can query and process the latest data record or the data within the tumbling window.
Data size	It processes large batches of data.	It processes individual records or micro batches consisting of a few records.
Performance	It achieves a processing delay of several minutes or hours.	It achieves a processing delay of several seconds and even milliseconds.
Analysis	You can perform complex data analysis.	You can perform simple analysis, such as simple response functions, aggregates, and rolling metrics.

Realtime Compute uses a simple computing model. Real-time computing of Realtime Compute makes significant improvements to batch computing in most scenarios of big data computing. In particular, in scenarios where event streams need to be processed with an extremely low processing delay, real-time computing is a valuable service for big data computing.

26.4. Benefits

Realtime Compute provides competitive advantages in stream processing, which allows you to easily perform real-time big data analysis. It offers the following benefits:

Powerful real-time computing functions

Realtime Compute simplifies the development process by integrating a wide range of features. Realtime Compute integrates the following features:

- A powerful engine is used. This engine offers the following advantages:
 - Provides standard Flink SQL that enables automatic data recovery from failures. This ensures accurate data processing if failures occur.
 - Supports a variety of built-in functions, such as string, date and time, and statistical functions.
 - Enables accurate control over computing resources. This ensures complete isolation of jobs of different tenants.
- Realtime Compute outperforms Apache Flink by three to four times when measured by key performance metrics. For example, in Realtime Compute, the data processing delay can be reduced to seconds or even to milliseconds. The throughput of a job can reach millions of data records per second, and a cluster can contain thousands of nodes.
- Realtime Compute integrates cloud-based data stores such as DataHub, Log Service, ApsaraDB RDS,

Tablestore, and AnalyticDB for MySQL. Realtime Compute can read data from and write data to these systems with the least efforts in data integration.

Managed real-time computing services

Unlike open source or self-managed stream processing services, Realtime Compute is a fully managed stream processing engine. You can query streaming data without the need to deploy or manage any infrastructure. Realtime Compute allows you to use streaming data processing services with a few clicks. Realtime Compute integrates services such as data development, data administration, monitoring, and alerting. This allows you to use cost-effective streaming data services for trial and migrate your data for deployment.

Realtime Compute also enables complete isolation between tenants. The isolation and protection extend from the top application layer to the underlying infrastructure layer. This helps ensure the security and privacy of your data.

- Isolation: allows the tasks of multiple tenants (projects) to be submitted to different queues and run separately. Resources are isolated among tenants.
- Permission: centrally manages different tenants to implement dynamic configurations and management of tenant resources, resource isolation, and usage statistics of resources. Management of multi-level tenants is supported.
- Scheduling: supports multi-tenant scheduling of multiple clusters and multiple resource pools.

You can use the cloud management platform to centrally manage user permissions. This platform assigns different permissions to different administrators for fine-grained permission control. The cloud management platform centrally displays and manages the permissions of each component, which simplifies permission management. Common users are unable to view details of permission management. Permission management operations for administrators are simplified. You can also manage data access permissions such as logon permissions on the cloud management platform.

Excellent user experience during development

Supports standard Flink SQL. Realtime Compute uses various built-in computing functions such as string, date and time, and statistical functions to simplify Flink SQL statements. This allows more business intelligence (BI) engineers and operational staff to perform real-time big data analysis and processing by using simple Flink SQL statements, which makes real-time big data processing universal.

Provides an end-to-end stream processing solution, including data development, data administration, monitoring, and alerting.

Supports automatic fault tolerance for server hard disk failures of a cluster, and supports hot swapping of hard disks. In the event of a hard disk failure, services can be restored within two minutes.

Adopts a multi-node cluster architecture. The management nodes in the platform support a high-availability mechanism. Faults on daily O&M management nodes do not affect normal business operations.

Low costs in labors and compute clusters

Alibaba Cloud has made many improvements to the SQL execution engine, which allows you to create jobs more cost-effectively than Flink jobs. Realtime Compute is more cost-effective than open source stream frameworks in both development and production costs. Realtime Compute allows you to fully focus on your business and quickly achieve your market goals without the need to write the Java code for Storm jobs with complex business logic. An Apache Storm job with complex business logic incurs high costs and demands a lot of effort to perform tasks such as writing enormous lines of Java code, debugging, testing, performance tuning, publishing, and long-term administration of open source software applications like Apache Storm and Zookeeper.

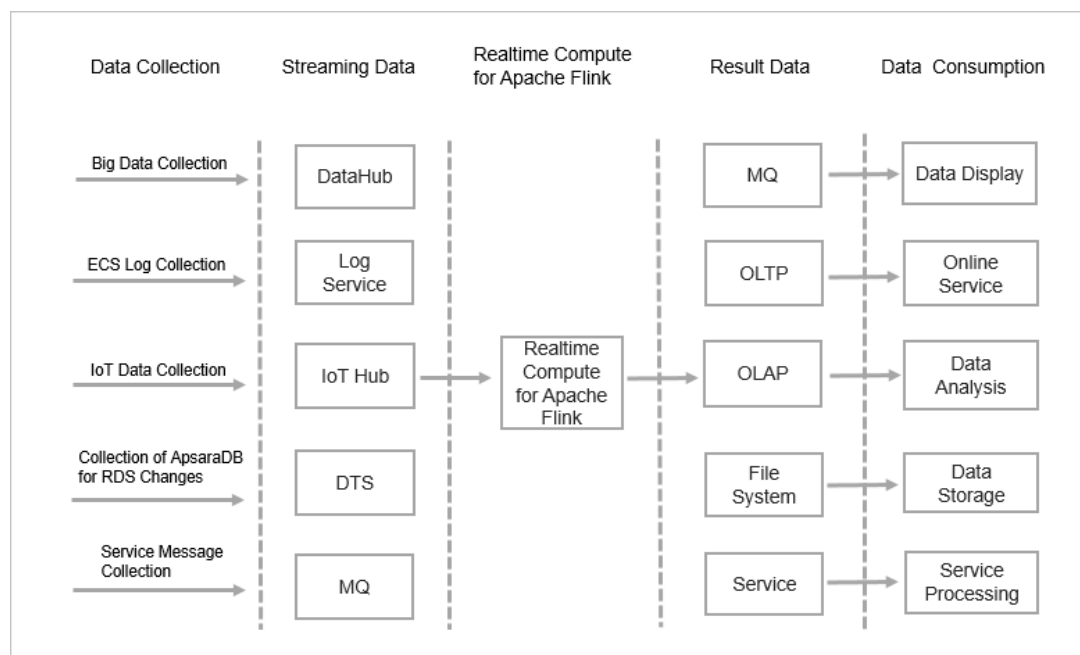
Supports CPUs, hard disks, memory, and NICs of different specifications in a single-component cluster without affecting cluster running performance. This guarantees maximum compatibility with existing devices.

26.5. Product architecture

26.5.1. Workflow

We recommend that you have a general knowledge about the stream processing architecture of Realtime Compute before you use this service. This helps you create effective plans for the design of stream processing systems. The following figure shows the architecture of Realtime Compute.

Architecture



- **Data collection**

You can use data collection tools to collect and send streaming data in real time to a publish-subscribe system for big data analysis. This publish-subscribe system continuously produces events for Realtime Compute in the downstream to trigger real-time computing jobs. The big data ecosystem of Alibaba Cloud offers a wide range of publish-subscribe systems to process streaming data in diversified scenarios. Realtime Compute integrates many of these systems, as shown in the preceding figure. This allows you to easily integrate multiple streaming data stores. To enable compatibility between the computing model of Realtime Compute and that of some data stores, another data store may be required for data processing. Realtime Compute is seamlessly connected to the following data stores:

- DataHub

DataHub allows you to upload data to its system by using a wide range of tools and on UIs. For example, you can upload logs, binary log files, and IoT streaming data to DataHub. DataHub also integrates open source business software applications. For more information about the data collection tools of DataHub, see DataHub documentation.

- Log Service

Log Service is an end-to-end logging service that is developed by Alibaba Group based on years of experience in addressing challenges involving large amounts of big data experienced by Alibaba Group. Log Service allows you to collect, transfer, query, consume, and analyze log data.

- IoT Hub

IoT Hub is a service that enables developers of IoT applications to implement two-way communications between devices (such as sensors, final control elements, embedded devices, and smart home appliances) and the cloud by creating secure data channels.

You can use the IoT Hub rules engine to send IoT data to DataHub, and use Realtime Compute and MaxCompute to process and perform computations on data.

- DTS

Data Transmission Service (DTS) supports data transmission between structured data stores represented by databases. DTS is a data exchange service that streamlines data migration, data synchronization, and data subscription. You can use the data transmission feature of DTS to parse binary log files such as ApsaraDB RDS logs and send data to DataHub. Realtime Compute and MaxCompute allow you to perform computations on the data.

- MQ

Message Queue (MQ) is a key service that provides messaging capabilities, such as message publishing and subscription, message tracing, scheduled and delayed messages, resource statistics, monitoring, and alerting. MQ offers a complete set of enterprise-level messaging features powered by high-availability (HA) distributed systems and clusters.

- Realtime Compute

Data streams continuously enter Realtime Compute for real-time computing. At least one data stream must enter Realtime Compute to trigger a real-time computing job. In complex business scenarios, Realtime Compute allows you to perform association queries for static data from data stores and streaming data. For example, you can perform JOIN operations on DataHub and ApsaraDB RDS tables based on the primary key of streaming data. You can then perform association queries on DataHub streaming data and ApsaraDB RDS static data. Realtime Compute also allows you to associate multiple data streams. Flink SQL allows you to handle large amounts of data and complex business scenarios, such as those experienced by Alibaba Group.

- Real-time data integration

To minimize the data processing delay and simplify data transmission links, Realtime Compute directly writes the result data of real-time computing to data sinks. Realtime Compute allows for a larger Alibaba Cloud ecosystem by integrating the following systems: online transaction processing (OLTP) systems such as ApsaraDB RDS, NoSQL database services such as Tablestore, online analytical processing (OLAP) systems such as AnalyticDB, message queue systems such as DataHub and MQ, and mass storage systems such as Object Storage Service (OSS) and MaxCompute.

- Data consumption

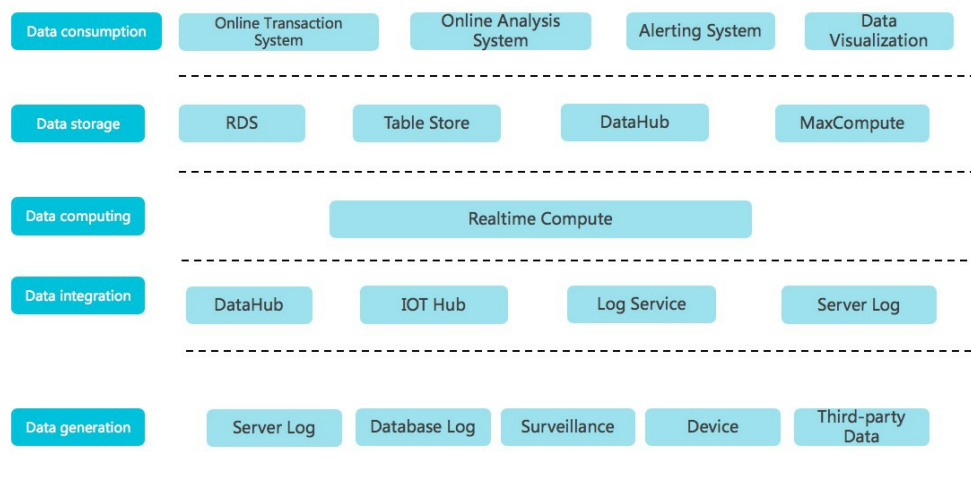
After the result data of real-time computing is written to the sinks, you can consume the data by using custom applications.

- Use data stores to access the result data.
- Use data transfer systems to send and receive the result data.
- Use alerting systems to send alerts.

26.5.2. Business architecture

Realtime Compute is a lightweight SQL-enabled streaming engine for real-time processing and analysis of data streams.

Business architecture



- Data generation

In this phase, streaming data is generated from sources such as server logs, database logs, sensors, and third-party systems. The generated streaming data moves on to the next phase for data integration to drive real-time computing.

- Data integration

In this phase, the streaming data is integrated. You can subscribe to and publish the integrated streaming data. The following Alibaba Cloud products can be used in this phase: DataHub for big data computing, IoT Hub for connecting IoT devices, and Log Service for integrating ECS logs.

- Data computing

In this phase, the streaming data, which has been subscribed to in the data integration phase, acts as inputs to drive real-time computing in Realtime Compute.

- Data store

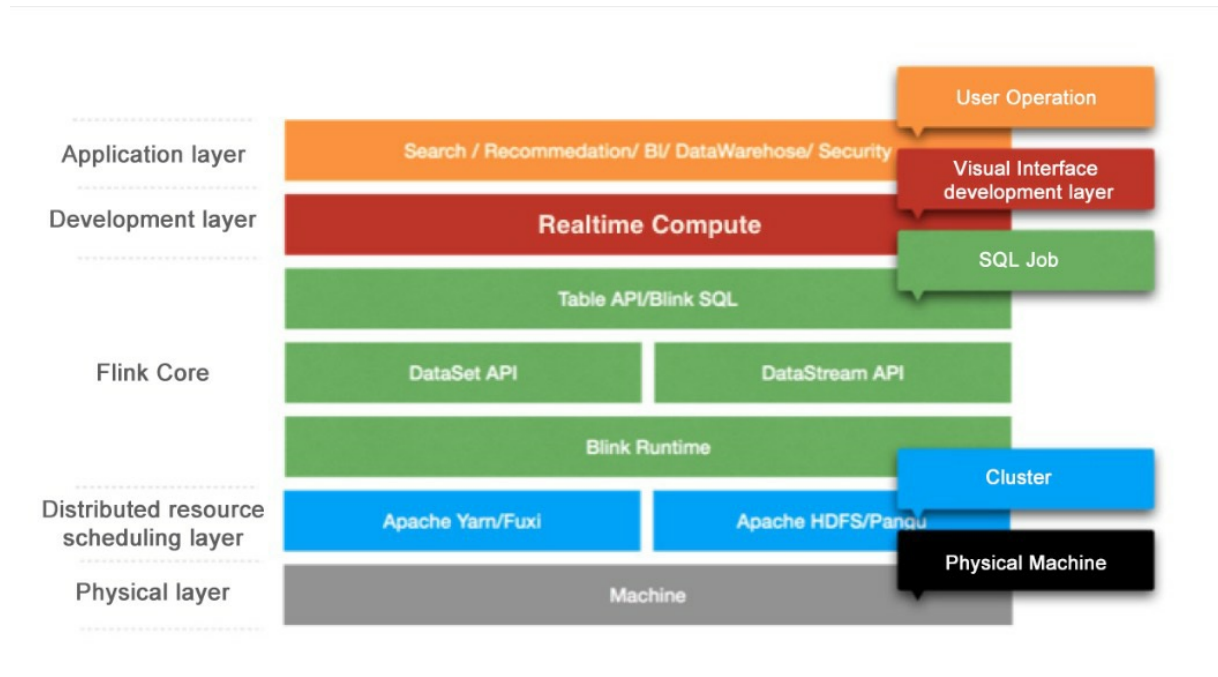
Realtime Compute does not provide built-in data stores. Instead, it writes computing results to external data stores, such as relational databases, NoSQL databases, and online analytical processing (OLAP) systems.

- Data consumption

Realtime Compute supports multiple data store types, which allows you to consume data in various ways. For example, data stores for message queues can be used to report alerts, and relational databases can be used to provide online support.

26.5.3. Technical architecture

Realtime Compute is a real-time data analysis platform for incremental computing. This platform provides statements that are similar to SQL statements and uses the MapReduceMerge (MRM) computing model for incremental computing. Realtime Compute offers a failover mechanism to ensure data accuracy when errors occur.



The Realtime Compute architecture consists of the following five layers.

- Application layer

This layer allows you to create SQL files and publish jobs for real-time data processing based on a development platform. With a well-designed monitoring and alerting system, you would be notified of a processing delay for each job in a timely manner. You can also use systems like Flink UI to view the running information of published jobs and analyze performance bottlenecks. This allows you to quickly and effectively improve job performance.

- Development layer

This layer parses Flink SQL and generates logical and physical execution plans. The execution plans are then conceptualized as executable directed acyclic graphs (DAGs). Based on these DAGs, directed graphs that consist of various models are obtained. Directed graphs are used to implement specific business logic. A model usually contains the following three modules:

- Map: Operations such as data filtering, distribution (GROUP), and join (MAPJOIN) are performed.
- Reduce: Realtime Compute processes streaming data by batch, and each batch contains multiple data records.

- Merge: You can update the state by merging the computing results of the batch, which are produced from the Reduce module, with the previous state. Checkpoints are created after N (configurable) batches have been processed. In this way, the state is stored persistently in a data store, such as Tair and Apache HBase.

- Flink Core

This layer provides a wide range of computing models, Table API, and Flink SQL. You can use DataStream API and DataSet API at the lower sublayer. At the bottom sublayer is Flink Runtime, which schedules resources to ensure that jobs can run properly.

- Distributed resource scheduling layer

Realtime Compute clusters run based on the Gallardo scheduling system. This system ensures that Realtime Compute runs effectively and fault tolerance is provided for recovery.

- Physical layer

This layer provides powerful hardware devices for clusters.

26.6. Features

Realtime Compute has the following features:

- **Data collection and storage**

Before you run a big data analysis system, you must ensure that data has been collected into the system. To make full use of your existing streaming data store, Realtime Compute can integrate with multiple upstream streaming data stores, such as DataHub, Log Service, IoT Hub, Tablestore, and Message Queue (MQ). You can use streaming data in existing data stores without the need to perform data collection and data integration.

You can register data stores on the Realtime Compute development platform. This enables you to use the advantages of the end-to-end Realtime Compute development platform. Realtime Compute provides UIs for managing different data stores, such as ApsaraDB RDS, AnalyticDB for MySQL, and Tablestore. Realtime Compute allows you to manage cloud-based data stores in one stop.

- **Data development**

- Realtime Compute provides a fully managed online development platform that integrates a wide range of Flink SQL coding assistance features, such as syntax check, automatic prompting, and syntax highlighting.

- **Syntax check**

On the Development page of Realtime Compute, the revised script is automatically saved. When the script is saved, an SQL syntax check is automatically performed. If a syntax error is detected, the Development page shows the row and column where the error is located, and the description of the error.

- **Automatic prompting**

When you enter SQL statements on the Development page of Realtime Compute, auto-completion prompts about keywords, built-in functions, and SQL statements are automatically displayed.

- **Syntax highlighting**

Flink SQL keywords are highlighted in different colors to differentiate data structures.

- The Realtime Compute development platform allows you to manage different versions of SQL code.

Realtime Compute provides key features that help you complete development tasks, such as coding assistance and code version management. On the data development platform, you can manage SQL code versions. The system generates a code version each time you commit code. The code version can be used for version tracking, modification, and rollback.

- Realtime Compute allows you to register data stores on the **Development** page for effective data store management, such as data preview and auto DDL generation.

- Preview data from a data store

The Development page of Realtime Compute allows you to preview the data of multiple data store types. Data preview helps you efficiently analyze upstream and downstream data, identify key business logic, and complete development tasks.

- Auto DDL generation

In most cases, the DDL statements for data stores are manually translated into the DDL statements for real-time computing. Therefore, the DDL generation process includes a large number of repetitive tasks. Realtime Compute provides an auto DDL generation feature. This feature simplifies the way that you edit SQL statements for stream processing jobs, reduces the possibility of encountering errors when you manually enter SQL statements, and also improves efficiency.

- Realtime Compute allows you to implement real-time data cleansing, statistics, and analysis by using standard SQL statements. Realtime Compute also supports common aggregate functions, and association queries for streaming data and static data.
- Realtime Compute provides a simulated running environment where you can customize uploaded data, simulate operations, and check output results.

- **Administration**

Realtime Compute allows you to manage stream processing jobs on the following tab pages under the Administration page: Overview, Curve Charts, FailOver, CheckPoints, JobManager, TaskExecutor, Data Lineage, and Properties and Parameters.

- **Performance optimization**

- Improve performance by automatic configuration

The automatic configuration feature of Realtime Compute helps you address performance issues, such as a low throughput of jobs or upstream and downstream backpressure.

- Improve performance by using manual configuration

You can manually configure resources to improve job performance by using one of the following methods:

- Optimize the resource configuration. You can modify the resources to improve performance by reconfiguring parameters, such as parallelism, core, and heap_memory.
- Improve performance based on job parameter settings. You can specify the job parameters such as miniBatch to improve performance.
- Improve upstream and downstream data stores based on parameter settings. You can specify related parameters to optimize the upstream and downstream data stores for a job.

- **Monitoring and alerting**

Cloud Monitor allows you to collect the performance metrics of cloud resources or other custom performance metrics, view service availability, and specify alerts based on the performance metrics. This way, you can view the cloud resource usage and running information of jobs. You can also receive and respond to alerts in a timely manner to ensure that applications can run properly. Realtime Compute allows you to specify alerts for the following performance metrics:

- Processing delay
- Input RPS
- Output RPS
- FailoverRate

26.7. Product positioning

Realtime Compute offers Flink SQL to support standard SQL semantics and help you easily implement the computational logic of stream processing. Realtime Compute also provides full-featured UDFs for some authorized users, helping you customize business-specific data processing logic in scenarios where SQL code functions cannot meet your business needs. In the field of streaming data analysis, you can directly use Flink SQL and UDFs to enable most of the streaming data analysis and processing logic. Realtime Compute focuses on the analysis, statistics, and processing of streaming data. It is less applicable to non-SQL businesses, such as complex iterative data processing and complex rule engine alerts.

Realtime Compute is applicable to the following scenarios:

- Collects the data about page views (PVs) and unique visitors (UVs) in real time.
- Collects the data about the average traffic flow at a traffic checkpoint every 5 minutes.
- Collects and displays the pressure data of hydroelectric dams.
- Reports alerts for financial thefts in online payment services based on fixed rules.

Realtime Compute is inapplicable to the following scenarios for now:

- Replacing Oracle stored procedures with Realtime Compute: Realtime Compute cannot implement all the functions of Oracle stored procedures, because they are designed to handle issues in different fields.
- Seamlessly migrating Spark jobs to Realtime Compute: Currently, you cannot seamlessly migrate Spark jobs to Realtime Compute. However, you can change the stream processing of Apache Spark and migrate this part to Realtime Compute. This eliminates various Apache Spark administration tasks and Spark-based development costs.
- Complex rule engines for alerting: Realtime Compute cannot handle scenarios where multiple complex alerting rules are specified for each data record, and the rules continue to change when the system is running. Specific rule engines need to be used to resolve these issues.

Realtime Compute provides a full set of development tools for streaming data analysis, statistics, and processing based on UDFs and Flink SQL. It allows you to devote the least efforts in developing the underlying code and simply write SQL statements to analyze streaming data. This makes Realtime Compute a good choice for users such as data warehouse developers and data analysts.

26.8. Scenarios

26.8.1. Overview

Realtime Compute uses Flink SQL to provide solutions for streaming data analysis.

- Real-time extract-transform-load (ETL)

Realtime Compute allows you to cleanse, aggregate, and sort streaming data in real time by leveraging the advantages of multiple data channels and flexible data processing capabilities of SQL. Realtime Compute serves as an effective supplement and optimization of offline data warehouses and provides a computing channel for real-time data transmission.

- Real-time reports

Realtime Compute allows you to collect and process streaming data, monitor performance metrics of the business, and view corresponding reports in real time. This enables real-time data administration.

- Monitoring and alerting

Realtime Compute allows you to monitor systems and analyze user behavior in real time, which helps to identify faults and risks in real time.

- Online systems

Realtime Compute allows you to run real-time computations over data streams and view performance metrics in real time. You can shift strategies for online systems in a timely fashion. Realtime Compute can be widely used in various content delivery and intelligent mobile push scenarios.

26.8.2. Management of e-commerce activities

Realtime Compute has evolved into a reliable stream processing platform from Alibaba Group's big data architecture in the e-commerce industry. Realtime Compute is suitable for analyzing various streaming data and providing report support in the e-commerce industry. The e-commerce industry needs to process streaming data in real time in the following scenarios:

- Real-time analysis of user behavior, for example, display of transaction data and user data on big screens. In traditional batch processing models, large amounts of data are processed inefficiently with a long delay. The size of the result data may be excessively large, which poses considerable challenges for online systems that are used for displaying the result data. This may compromise the stability of the online systems.
- Real-time monitoring of users, services, and systems. For example, marketers and engineers can have knowledge of the transactions on the platform over a specified period by viewing the corresponding curve chart. If abnormal fluctuations occur, such as a sharp decrease in transactions, alerts must be instantly triggered and sent to users. This helps users effectively respond to abnormal fluctuations and reduces negative impacts on the business.
- Real-time monitoring of major promotional events. For example, marketers need to monitor the metrics of promotional events in real time, such as the Double 11 Shopping Festival created by Alibaba Group and 618 mid-year shopping festival started by JD.com, Inc. This helps marketers effectively decide whether to change strategies.

Integrating with Alibaba Cloud computing and storage systems, Realtime Compute allows you to meet your custom needs for streaming data analysis. Realtime Compute not only satisfies diverse business needs but also simplifies the business development process by using Flink SQL.

26.8.3. Multidimensional analysis of data from IoT sensors

Background

With the economic tidal wave of globalization sweeping over the world, industrial manufacturers are facing increasingly fierce competition. To increase competitiveness, manufacturers in the automotive, aviation, high-tech, food and beverage, textile, and pharmaceutical industries must innovate and replace the existing infrastructure. These industries have to address many challenges during the innovation process. For example, the existing traditional devices and systems have been used for decades, which results in high maintenance costs. However, replacing these systems and devices may slow down the production process and compromise the product quality.

These industries face two additional challenges, which are high security risks and the urgent need for complex process automation. The manufacturing industry has prepared to replace the existing traditional devices and systems. In this industry, high reliability and availability systems are needed to ensure the safety and stability of real-time operations. A manufacturing process involves a wide range of components, such as robotic arms, assembly lines, and packaging machines. This requires remote applications that can seamlessly integrate each stage of the manufacturing process, including the deployment, update, and end-of-life management of devices. The remote applications also need to handle failover issues.

Another requirement for these next-generation systems and applications is that they be able to capture and analyze the large amounts of data generated by devices, and respond appropriately in a timely manner. To increase competitiveness and accelerate development, manufacturers need to optimize and upgrade their existing systems and devices. The application of Realtime Compute and Alibaba Cloud IoT solutions allows you to analyze device running information, detect faults, and predict yield rates in real time. This topic describes a use case as an example. In this use case, a manufacturer uses Realtime Compute to analyze the large amounts of data collected from sensors in real time. Realtime Compute is also used to cleanse and aggregate data in real time, write data to an online analytical processing (OLAP) system in real time, and monitor the key metrics of devices in real time.

Scenario description

In this use case, the manufacturer has more than 1,000 devices from multiple factories in many cities. Each device is equipped with 10 types of sensors. These sensors send the collected data every 5 seconds to Log Service. The data collected from each sensor follows the format described in the following table.

s_id	s_value	s_ts
The ID of the sensor.	The current value from the sensor.	The time when the data was sent.

The sensors are distributed across devices from multiple factories. The manufacturer creates an RDS dimension table to display the distribution of sensors across devices and factories.

s_id	s_type	device_id	factory_id
The ID of the sensor.	The type of the sensor.	The ID of the device.	The ID of the factory.

The information included in this dimension table is stored in the RDS system. The manufacturer needs to organize the data from sensors based on this dimension table, and sort the data by device. To meet this need, Realtime Compute provides a summary table where the data sent from sensors is logically aggregated by device every minute.

ts	device_id	factory_id	device_temp	device_pres
The time when the data was sent.	The ID of the device.	The ID of the factory.	The temperature of the device.	The pressure of the device.

Assume that there are only two types of sensors in this use case: temperature and pressure. The computational logic is described as follows:

1. Realtime Compute identifies the devices whose temperatures are higher than 80°C and triggers alerts at the downstream nodes. In this use case, Realtime Compute sends the data of the identified devices to MQ. MQ then triggers alerts that the manufacturer has specified in the downstream alerting system.
2. Realtime Compute writes the data to an OLAP system. In this use case, the manufacturer uses HybridDB for MySQL. To integrate with HybridDB for MySQL, the manufacturer has developed a set of business intelligence (BI) applications for multidimensional data display.

FAQ

- How can I aggregate data into a summary table?

In most cases, each sensor only collects the IoT data of one dimension. This poses challenges for subsequent data processing and analysis. To create a summary table, Realtime Compute aggregates data based on windows and organizes data by dimension.

- Why is MQ used to trigger alerts?

Realtime Compute allows you to write data to any type of storage system. We recommend that you use message storage systems like MQ for sending alerts and notifications. This is because the application of these systems helps to prevent the errors encountered by user-defined alerting systems. These errors may cause failures to report certain alerts and notifications.

Code description

Send the data uploaded from sensors to Log Service. The data format of a row is shown as follows:

```
{
  "sid": "t_xxsfdsad",
  "s_value": "85.5",
  "s_ts": "1515228763"
}
```

Define a Log Service source table `s_sensor_data`.

```
CREATE TABLE s_sensor_data (  
  s_id VARCHAR,  
  s_value VARCHAR,  
  s_ts VARCHAR,  
  ts AS CAST(FROM_UNIXTIME(CAST(s_ts AS BIGINT)) AS TIMESTAMP),  
  WATERMARK FOR ts AS withOffset(ts, 10000)  
) WITH (  
  TYPE='sls',  
  endPoint='http://cn-hangzhou-corp.sls.aliyuncs.com',  
  accessId='*****',  
  accessKey='*****',  
  project='ali-cloud-streamtest',  
  logStore='stream-test',  
);
```

Create an RDS dimension table `d_sensor_device_data`. This dimension table stores the mappings between sensors and devices.

```
CREATE TABLE d_sensor_device_data (  
  s_id VARCHAR,  
  s_type VARCHAR,  
  device_id BIGINT,  
  factory_id BIGINT,  
  PRIMARY KEY(s_id)  
) WITH (  
  TYPE='RDS',  
  url='',  
  tableName='test4',  
  userName='test',  
  password='*****'  
);
```

Create an MQ result table `r_monitor_data`. This table specifies the logic for triggering alerts.

```
CREATE TABLE r_monitor_data (  
  ts VARCHAR,  
  device_id BIGINT,  
  factory_id BIGINT,  
  device_TEMP DOUBLE,  
  device_PRES DOUBLE  
) WITH (  
  TYPE='MQ'  
);
```

Create a HybridDB for MySQL result table `r_device_data`.

```
CREATE TABLE r_device_data (
  ts VARCHAR,
  device_id BIGINT,
  factory_id BIGINT,
  device_temp DOUBLE,
  device_pres DOUBLE,
  PRIMARY KEY(ts, device_id)
) WITH (
  TYPE='HybridDB'
);
```

Aggregate the data collected from sensors by minute and create a summary table based on the aggregated data. To clearly view the code structure and facilitate subsequent administration, we create views in this use case.

```
// Create a view to obtain the device and factory mapping each sensor.
CREATE VIEW v_sensor_device_data
AS
SELECT
  s.ts,
  s.s_id,
  s.s_value,
  s.s_type,
  s.device_id,
  s.factory_id
FROM
  s_sensor_data s
JOIN
  d_sensor_device_data d
ON
  s.s_id = d.s_id;
// Aggregate the data collected from sensors.
CREATE VIEW v_device_data
AS
SELECT
  // Specify the start time of a tumbling window as the time for the record.
  CAST(TUMBLE_START(v.ts, INTERVAL '1' MINUTE) AS VARCHAR) as ts,
  v.device_id,
  v.factory_id,
  CAST(SUM(IF(v.s_type = 'TEMP', v.s_value, 0)) AS DOUBLE)/CAST(SUM(IF(v.s_type = 'TEMP', 1, 0)) AS DOUBLE) device_temp, // Compute the average temperature by minute.
  CAST(SUM(IF(v.s_type = 'PRES', v.s_value, 0)) AS DOUBLE)/CAST(SUM(IF(v.s_type = 'PRES', 1, 0)) AS DOUBLE) device_pres // Compute the average pressure by minute.
FROM
  v_sensor_device_data v
GROUP BY
  TUMBLE(v.ts, INTERVAL '1' MINUTE), v.device_id, v.factory_id;
```

In the preceding core computational logic, the average temperature and pressure by minute are computed as the output. Tumbling windows are used in this use case. A new window is started every minute, and a new set of data is generated every minute. The generated data is then filtered and written to the MQ result table and HybridDB result table.

```
// Identify the sensors whose temperatures are higher than 80°C and write the data to the MQ result table to trigger alerts.
INSERT INTO r_monitor_data
SELECT
    ts,
    device_id,
    factory_id,
    device_temp,
    device_pres
FROM
    v_device_data
WHERE
    device_temp > 80.0;
// Write the result data to the HybridDB for MySQL result table for analysis.
INSERT INTO r_device_data
SELECT
    ts,
    device_id,
    factory_id,
    device_temp,
    device_pres
FROM
    v_device_data;
```

26.8.4. Big screen service for the Tmall Double 11 Shopping Festival

The annual Tmall Double 11 Shopping Festival has become the largest sales event for online shopping in the world. A large number of netizens demonstrate a strong desire to purchase products during the sales event each year. One of the key highlights of this event has been the increase in the overall turnover that is displayed on the Tmall big screen in real time. The real-time display of turnover on the big screen is a result of our senior engineers' hard work over several months. The big screen service excels in key performance metrics. For example, the end-to-end delay has been reduced within 5 seconds, from placing orders on the Tmall platform, to data collection, processing, verification, and to displaying the sales data on the big screen. As for the processing capability, hundreds of thousands of orders can be processed during the peak hours around 00:00 on November 11. Additionally, to ensure fault tolerance, multiple channels have been used to back up data.

Realtime Compute provides key support for the big screen service. The stream processing of the big screen service was previously based on the open source Apache Storm. The Storm-based development process took around one month. The application of Flink SQL shortened the development process of the big screen service to one week. The underlying layer of Realtime Compute removes the Apache Storm modules that are designed for execution optimization and troubleshooting. This enables higher efficiency and faster processing for Realtime Compute jobs.

- Online shopping rush

During the Double 11 Shopping Festival, an enormous number of netizens join the online shopping rush on the Tmall platform. During the peak hours when "seckill" activities occur, such as 00:00 on November 11, hundreds of thousands of sales orders need to be processed in real time. The word "seckill" vividly describes fighting among buyers, which means that a buyer wins or loses all in a matter of seconds.

- Real-time data collection

The data collection system collects and sends the logs of database changes to the DataHub system. With the application of Data Transmission Service (DTS), the data from online transaction processing databases can be written to DataHub tables within seconds at the peak hours around 00:00 on November 11.

- Real-time data computing

Realtime Compute subscribes to the DataHub streaming data, continuously analyzes the streaming data, and calculates the total turnover up to the current time. In Realtime Compute, a cluster can contain up to thousands of nodes. The throughput of a job reaches millions of data records per second, fully meeting the system requirements of processing hundreds of thousands of transactions per second in Tmall. Realtime Compute subscribes to data and writes the result data to an online RDS system in real time.

- Frontend data visualization

We also provide advanced data visualization components for the Tmall Double 11 Shopping Festival. These components allow you to view the total turnover on a dashboard, and the distribution of global transaction activities across the world in real time. To achieve astounding visual effects for the big screen, the frontend server performs periodic polling operations on the RDS system, and advanced web frontend applications are used.

26.8.5. Mobile data analysis

Realtime Compute allows you to analyze the data of mobile apps in real time. It helps you analyze the performance metrics of mobile apps, such as failure detection and distribution, and distribution of app versions. Mobile Analytics is a product provided by Alibaba Cloud to analyze the data of mobile apps. This product allows you to analyze user behavior and logs from multiple dimensions. It also helps mobile developers implement fine-grained operations based on big data analysis, improve product quality and customer experience, and enhance customer stickiness. The underlying big data computing of Mobile Analytics is implemented based on big data products of Alibaba Cloud, such as Realtime Compute and MaxCompute. Mobile Analytics uses Realtime Compute as the underlying engine for streaming data analysis. This allows Mobile Analytics to offer a wide range of real-time data analysis and reporting services for mobile apps.

- Data collection

To collect data, developers can include the SDK provided by Mobile Analytics into an app installation package. This SDK offers data collection components based on mobile operating systems. These components collect and send the data about mobile phones and user behavior to the backend of Mobile Analytics for analysis.

- Data reporting

The backend of Mobile Analytics offers a data reporting system, which allows you to collect the data reported by mobile phones by using the specified SDK. The data reporting system preliminarily removes dirty data and sends the processed data to DataHub.

 **Note** In the future, DataHub will provide an SDK for mobile phones to directly report data. The removal of dirty data will then be performed in Realtime Compute instead of Mobile Analytics, reducing the host costs of Mobile Analytics.

- Stream processing

Realtime Compute continuously subscribes to the DataHub streaming data. It also continuously reads and runs computations over the data about the performance metrics of mobile apps. Realtime Compute then writes the result data of stream processing during each period to an online ApsaraDB RDS or Tablestore system.

- Data display

Mobile Analytics provides a complete set of performance metrics that allow you to view the running information and usage of mobile apps. For example, you can know user locations, visited pages, browsing duration, end devices and network environments, and slow responses or failures. Mobile Analytics also allows you to analyze failures by device and view the details of failures. The data display is based on the result data that is obtained in the stream processing phase.

26.9. Restrictions

None

26.10. Basic concepts

Project

In Realtime Compute, a project is a basic unit for managing clusters, jobs, resources, and users. Project administrators can create projects, or add users to other existing projects. Realtime Compute projects can be collaboratively managed by Apsara Stack tenant accounts and RAM users.

Job

Similar to a MaxCompute or Hadoop job, a Realtime Compute job implements the computational logic of stream processing. A job is a basic unit for stream processing.

CU

In Realtime Compute, a compute unit (CU) defines the minimum capabilities of stream processing for a job with the specified CPU cores, memory, and I/O capacities. A Realtime Compute job can use one or more CUs.

A CU is assigned with one CPU core and 4 GB of memory .

Flink SQL

Unlike most open source stream processing systems that provide basic APIs, Realtime Compute offers Flink SQL that includes standard SQL semantics and advanced semantics for stream processing. Flink SQL is designed to satisfy diversified business needs, and it allows developers to perform stream processing by using standard SQL. Even users with limited technological skills, such as data analysts, can quickly and easily process and analyze streaming data by using Realtime Compute, .

UDF

Realtime Compute allows you to use user-defined functions (UDFs) that are similar to Apache Hive UDFs. We recommend that you use UDFs to implement your custom computational logic. UDFs are a supplement to Flink SQL that can be used for standard stream processing. Realtime Compute supports UDFs in Java (recommended) and Python.

Resource

Realtime Compute supports UDFs in Java (recommended) and Python. A JAR file uploaded by a user is defined as a resource.

Data collection

During a typical data collection process, data is collected from sources and ingested into a big data processing engine. The data collection process of Realtime Compute focuses on the phases where data is collected from the source and then transferred into a data bus.

Data store

Realtime Compute is a lightweight computing engine without built-in data stores. Data sources and sinks of Realtime Compute are based on external data stores. For example, you can use Apsara RDS to store result tables for Realtime Compute.

Data development

During the data development process, you can edit Flink SQL statements to create a Realtime Compute job. Realtime Compute offers an online integrated development environment (IDE) where you can edit SQL statements and debug data before you publish a Realtime Compute job.

Data administration

The data administration page of Realtime Compute allows you to manage jobs online. Realtime Compute helps you easily and effectively manage stream processing jobs.

27. Machine Learning Platform for AI

27.1. What is Machine Learning Platform for AI?

Machine Learning Platform for AI uses statistical algorithms to train models by using a large amount of historical data and generate empirical models. You can use these models to make informed business decisions.

Machine Learning Platform for AI is a set of tools for data mining, modeling, and prediction. It is developed based on the distributed computing engine MaxCompute. Machine Learning Platform for AI provides you with the following benefits:

- Machine Learning Platform for AI provides you with an all-in-one algorithm service that supports algorithm development, sharing, model training, deployment, and monitoring.
- Machine Learning Platform for AI allows you to manage experiments on the graphic user interface (GUI) or by running commands. It is intended for data miners, analysts, algorithm developers, and data explorers.
- In Apsara Stack, Machine Learning Platform for AI runs on MaxCompute. After you deploy algorithms in MaxCompute clusters, you can call the algorithms from the Machine Learning Platform for AI console. This decouples algorithm applications from computing engines.
- Machine Learning Platform for AI provides you with bountiful algorithms and reliable technical support for you to resolve issues in various business scenarios. In the data technology (DT) era, you can use Machine Learning Platform for AI to develop data-driven business.

You can apply Machine Learning Platform for AI in the following scenarios:

- Marketing: commodity recommendation, user profiling, and targeted advertising
- Finance: credit risk prediction for loans, financial risk management, stock forecast, and gold price forecast
- Social network: analytics of key opinion leaders and relational networks
- Text processing: news classification, keyword extraction, document summarization, and text analysis
- Unstructured data processing: image classification and image text extraction based on Optical Character Recognition (OCR)
- Other scenarios: rainfall forecast and forecast of football match results

Machine Learning Platform for AI supports the following learning modes:

- Supervised learning: Each sample has an expected value. You can create a model to map input feature vectors to goal values. Typical examples of this learning mode include regression and classification.
- Unsupervised learning: Goal values are not specified for samples. This learning mode is used to discover potential principles based on the sample data, such as simple clustering.
- Reinforcement learning: This learning mode is complex. A system constantly interacts with external environments to obtain feedback and determines its own behaviors to achieve long-term optimization of objectives. Typical examples of this learning mode contain AlphaGo and autonomous vehicles.

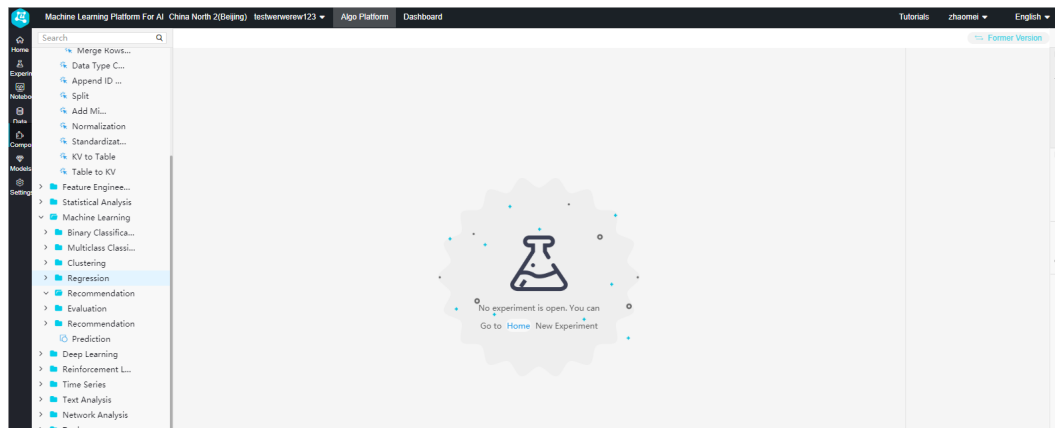
27.2. Benefits

Alibaba Cloud Machine Learning Platform for AI has the following benefits:

All-in-one visual user interface

- Machine Learning Platform for AI provides a Web interface for you to mine data by dragging and dropping components without programming, like piling up blocks, as shown in [User interface](#).

User interface



- Machine Learning Platform for AI provides the data model visualization function. It allows you to use charts to view data analysis results and algorithm evaluation.
- Machine Learning Platform for AI provides an all-in-one solution for data processing, model training, prediction, evaluation, model deployment, service building, and task scheduling.
- In addition to the Web interface, Machine Learning Platform for AI also provides command line tools to easily integrate algorithms into your projects.

Multiple high-performance machine learning algorithms

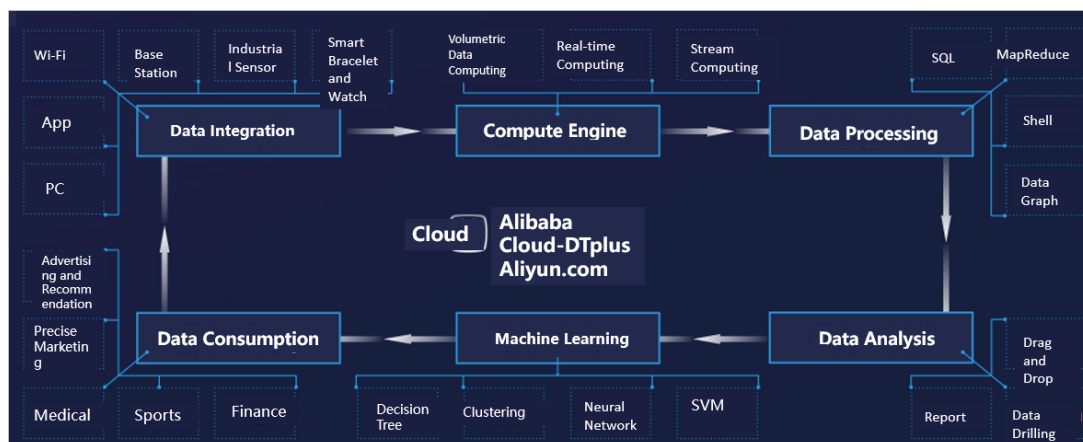
- Machine Learning Platform for AI provides nearly 100 machine learning algorithms that can be applied to multiple business scenarios, such as data preprocessing, clustering, regression, text analysis, and feature processing algorithms.
- Compared with traditional software, Machine Learning Platform for AI adopts the latest and optimal algorithms in the machine learning industry to improve the computing capability and accuracy.
- Machine Learning Platform for AI supports deep learning and GPU job scheduling. Machine Learning Platform for AI integrates and completely optimizes the TensorFlow framework. You can get started with TensorFlow for model training.
- Machine Learning Platform for AI provides open-source algorithms that are developed based on years of experience of Alibaba Cloud in big data mining and utilization. This greatly shortens the data modeling, model deployment, and model utilization period.

Full compatibility with Alibaba Cloud services

- Apsara Stack has established a big data ecosystem, such as Machine Learning Platform for AI. All services are ready for use after you activate them.
- Machine Learning Platform for AI runs on MaxCompute and is integrated with DTP DataWorks to help data mining, parent and child node data collection, experiment scheduling, and data utilization, as shown in [Alibaba Cloud DTP services](#).

- Based on the MPI, PS, graph algorithms, and MapReduce computing frameworks and distributed algorithms, Machine Learning Platform for AI easily handles a large amount of data.

Alibaba Cloud DTplus services



High-quality technical support

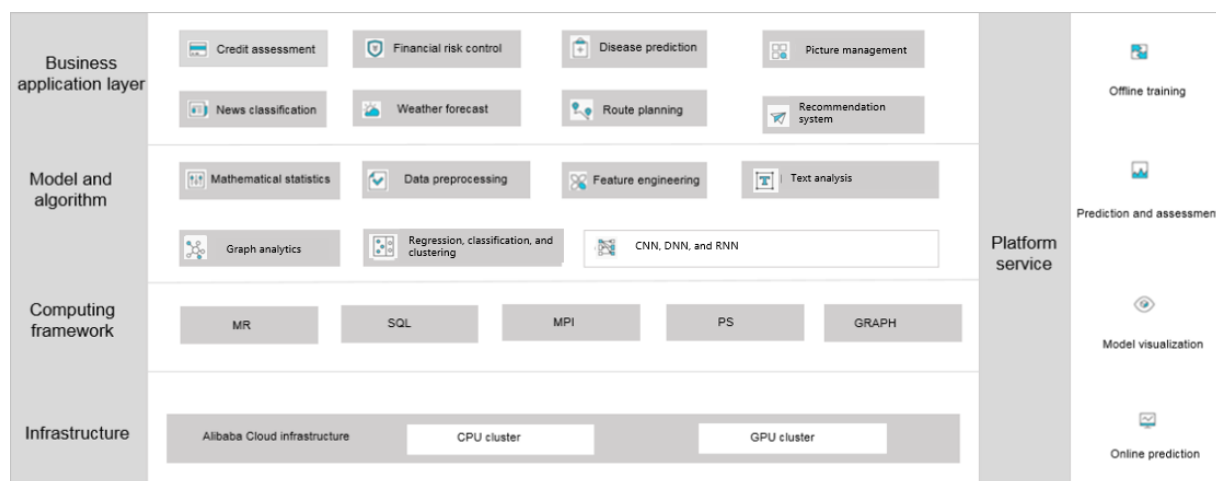
Machine Learning Platform for AI is supported by Alibaba algorithm scientists and Apsara Stack technical support. If you have any issues, submit a ticket through the ticket system or contact Apsara Stack technical personnel.

27.3. Architectures

This topic describes the architectures of Machine Learning Studio and Data Science Workshop (DSW) that are provided by Machine Learning Platform for AI (PAI).

Architecture of Machine Learning Studio

The following figure shows the architecture of Machine Learning Studio.



The architecture consists of the following layers:

- Infrastructure layer: provides cluster resources for computing. You can choose CPU or GPU clusters based on the algorithm type.
 - The CPU cluster runs machine learning algorithm jobs and provides CPU and memory resources for computing. Computing resources are centrally managed by an algorithm framework. After jobs are submitted, the algorithm framework distributes the jobs to the compute nodes in the CPU cluster.

- The GPU cluster runs deep learning framework jobs and provides GPU and memory resources for computing. Computing resources are centrally managed by an algorithm framework. After jobs are submitted, the algorithm framework distributes the jobs to the compute nodes in the GPU cluster. For a task that requires multiple multi-GPU servers, a virtual network is automatically created to distribute the jobs to the compute nodes in the virtual network.
- Computing framework layer: manages CPU and GPU resources and provides a basic runtime environment for algorithms. Supported runtime environments include the MapReduce runtime library, message passing interface (MPI) runtime library, parameter server (PS) runtime library, and TensorFlow framework.

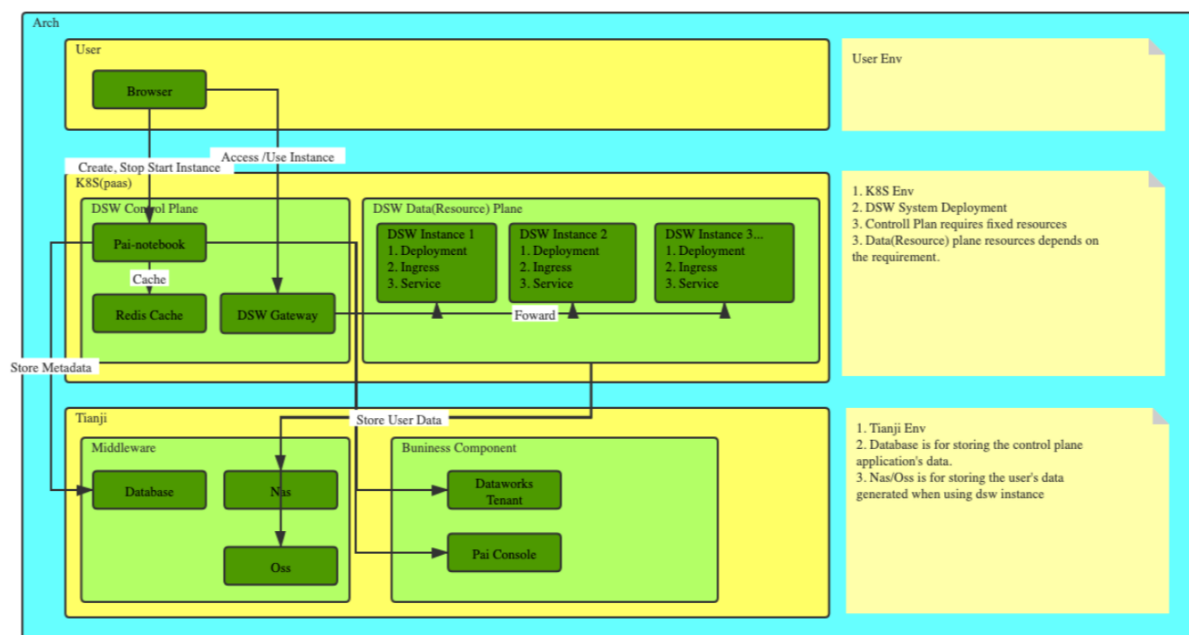
The open source version 1.4 of the deep learning framework TensorFlow is supported. PAI has optimized the performance of the open source version and improved its I/O interfaces. You can use TensorFlow to read objects from and write models to Object Storage Service (OSS) buckets. When TensorFlow is running, you can start TensorBoard to display the status of parameter convergence during convolution.

- Model and algorithm layer: provides basic components such as data preprocessing, feature engineering, and machine learning algorithms. All algorithm components are developed based on the algorithm system of Alibaba Group and have been tested by using petabytes of service data.
- Application layer: supports different Alibaba projects, such as the search system, recommendation system, and Ant Financial, in data mining. PAI is applicable in various industries, such as finance, medical care, education, transportation, and security.

On the basis of this architecture, PAI provides the web-based console GUI for you to manage algorithm experiments in a visualized manner. The GUI allows you to perform offline training, prediction, and evaluation. The GUI also allows you to visualize models, deploy online prediction services, or release experiments to the scheduling system of DataWorks.

Architecture of DSW

DSW of PAI provides an interactive environment for modeling. The architecture of DSW consists of the control plane and data plane, as shown in the following figure.



The architecture consists of the following layers:

- Control plane

The control plane includes DSW control services, as shown in the preceding figure. You can view resource lists and perform operations on resources in the control plane. For example, you can create, modify, and delete resources and query the information about resources in the control plane. The control plane provides API operations that allow you to manipulate the data plane. For example, it provides API operations for you to start pods, deploy networks, and create Apsara File Storage NAS (NAS) resources and mount targets for Kubernetes clusters.

- Data plane

The data plane includes Kubernetes clusters. Most of the DSW resources in the data plane belong to the dsw namespace. You can start pods to obtain the resources that are required to develop and run computing tasks.

27.4. Features

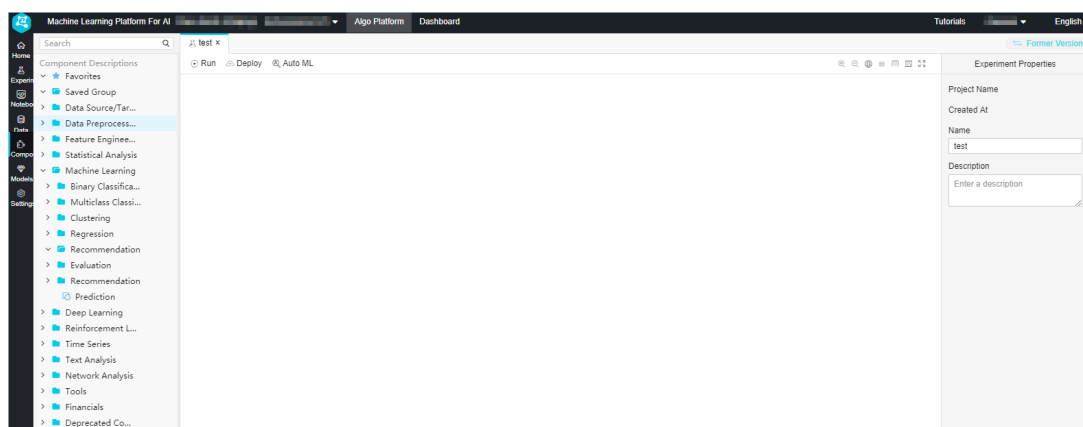
27.4.1. Visualized modeling

Machine Learning Platform for AI provides the easy-to-use visual modeling feature, which allows you to view the logic of the procedure.

The visual modeling pages include the algorithm platform page and online model service page.

[Algorithm platform page](#) shows the function section on the algorithm platform page. [Online model service page](#) shows the online model service page.

Algorithm platform page



Sections on the algorithm platform page

The algorithm platform page includes the following sections:

- Features section: displays machine learning features and information, such as experiments, components, data sources, and models, in a tree structure.
- Canvas section: You can drag and drop components to the canvas to build a directional workflow in order to complete data mining tasks, such as the metadata collection, data processing, modeling, and model deployment.
- Properties section: You can configure component parameters in this section.

Features section

The features section on the algorithm platform page provides the following menus:

- Search: You can search data, tables, and experiments.
- Experiments: After you double-click the name of an experiment, the canvas displays the directional flowchart of the experiment. You can continue modifying the experiment.
- Data sources: allows you to view and manage all data tables.
- Components: provides multiple key features of machine learning, such as machine learning components.
- Models: allows you to manage all models.
- Developer tool: allows you to view the experiment runtime log and troubleshoot experiment issues based on returned error messages and alerts.

Online model service

In the upper section of the page, select Online Model Service to go to the online model service page. This page displays user-created online services. You can monitor or select an action for these algorithm services.

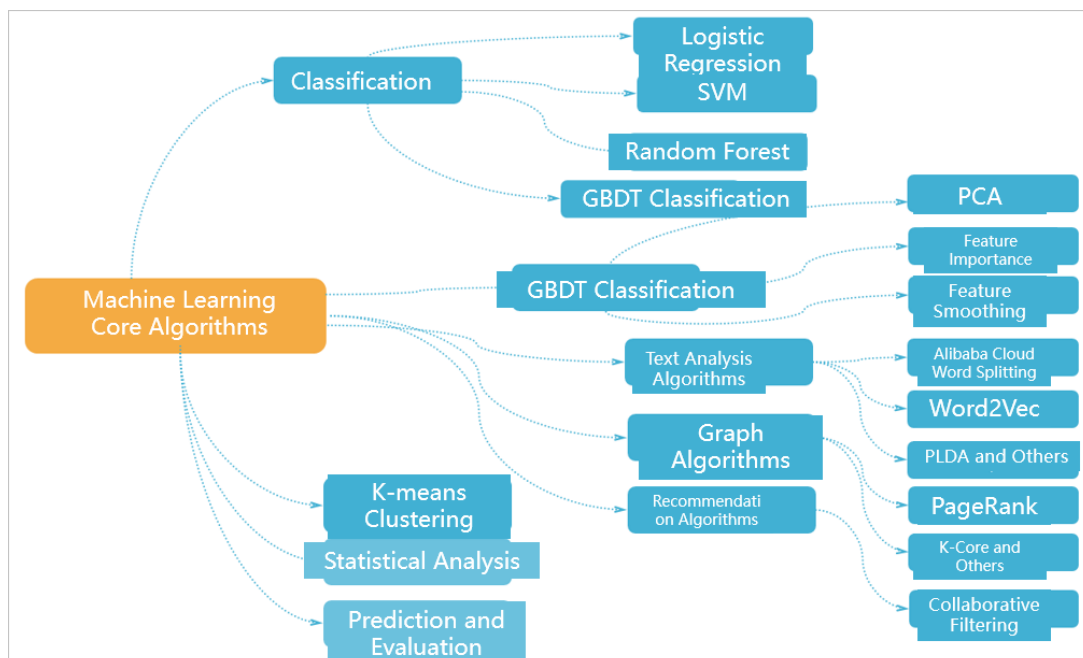
Online model service page

Model ID	Model Name	Created By	Monitor	Current Version	Status	Resources Used	Runtime	Updated At	Source	Actions
10	test1234			v1.0.0	Running	1Process 10Quota	97758"	11/28/2019 14:19:31		Update Delete Distribute Traffic

27.4.2. All-in-one experience

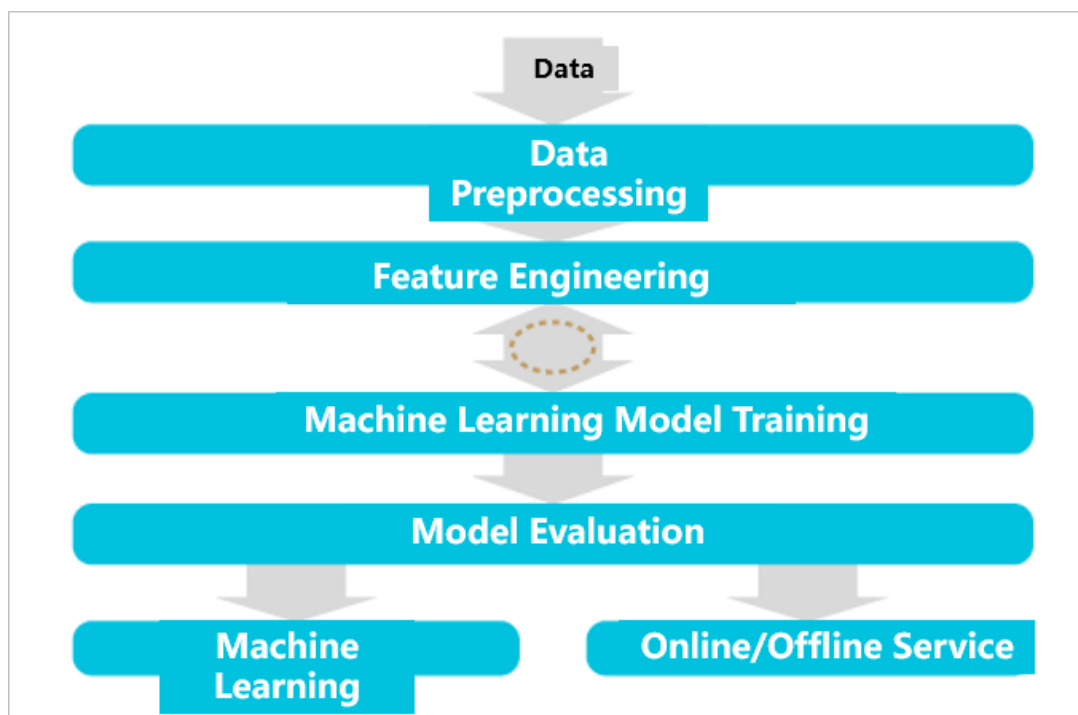
Machine Learning Platform for AI has a complete algorithm library, as shown in **Algorithm library of Machine Learning Platform for AI**.

Algorithm library of Machine Learning Platform for AI



Typically, you need to perform many operations to complete data mining or model training such as data extract, transform, and load (ETL), data preprocessing, feature engineering, model training, evaluation, and deployment, as shown in [Algorithm development process of atypical model](#). Machine Learning Platform for AI provides an all-in-one development environment with a complete set of components and tools for you to complete the entire data mining or model training task, such as metadata processing and model deployment. With these basic components, you can import data to the platform, create an experiment, and create solutions to resolve issues in different scenarios, and save costs on environment switching.

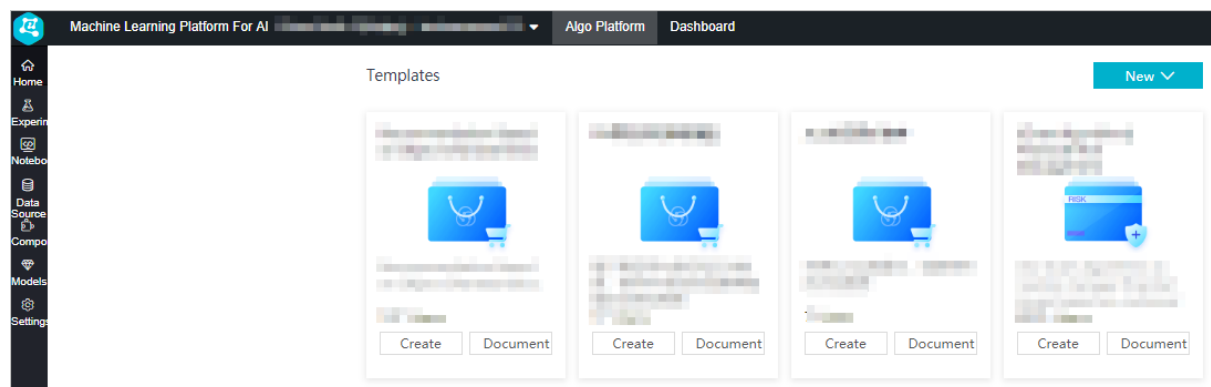
Procedure of developing algorithms for typical machine learning models



27.4.3. Multiple templates on the homepage

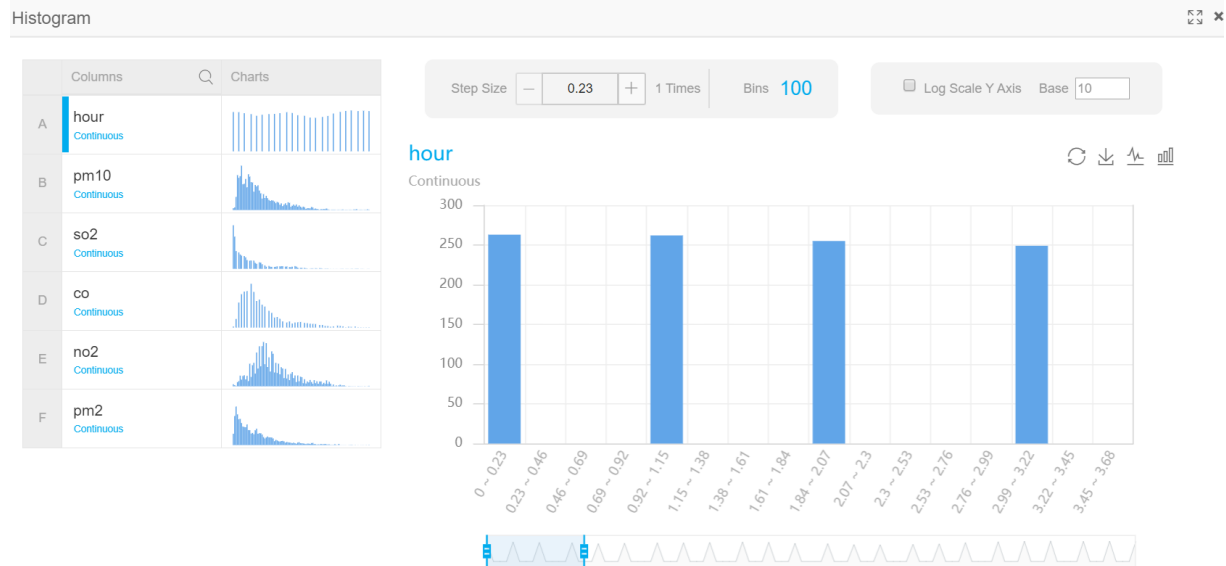
To help data analysts get started with the service, Machine Learning Platform for AI provides a set of experiment templates for scenarios such as product recommendations, text analysis, financial risk management, and weather prediction. These templates contain configurations and data that you can run the experiment with.

You can create experiments from the templates provided on the homepage. You can learn information about how an experiment is configured, how machine learning works, and how data is processed.



27.4.4. Data visualization

You can right-click an output component to view the visual output model. For example, you can view the model evaluation report and data analysis results. Visualized output can be displayed in multiple forms such as line charts, dot charts, and bar charts.



27.4.5. Model management

Visualized model management:

1. In the left-side navigation pane, click **Models**.
2. Expand the **Models** folder to locate the model built from a specified experiment.
3. Right-click the model and then select **Show Model**.

Binary Logistic Regression

If the input is sparse data, feature columns with zero weight are not displayed.

Column Name ▲	Weight	
	1 ▲	0 ▲
pm10	15.7360985405929	-
so2	1.474706135267199	-
co	0.8739231801649705	-
no2	8.652489608506198	-
Constants	-14.47551218333226	0

Save to MaxCompute: pai_lr_coefficient_xlab_m_logisticregres_1165164_v0 **Save** **Close**

 **Note** You can also right-click the model to perform other actions. For example, you can export the PMML file or deploy the model.

27.4.6. Algorithm components

Machine Learning Platform for AI (PAI) on Apsara Stack provides up to 89 algorithm components. These components are classified into 12 categories. The following table describes the algorithm components.

Level 1	Level 2	Level 3	Description
1.Data Source/Target	1.1 Read MaxCompute Table	-	Reads data from a MaxCompute table.
	1.2 Write MaxCompute Table	-	Writes data to a MaxCompute table.
	1.3 Read File Data	-	Reads data from Object Storage Service (OSS). The output of this component can be used only as the input of a deep learning or video-based intelligent algorithm component.
	1.4 MaxCompute TypeConverter	-	
2.Data Preprocessing	2.1 Sampling and Filtering	2.1.1 Weighted Sampling	Collects samples based on the specified ratio.
		2.1.2 Random Sampling	Randomly collects samples.
		2.1.3 Filtering and Mapping	Uses the SQL WHERE clause to filter data.
		2.1.4 Stratified Sampling	Collects samples by stratum.
	2.2 Data Merge	2.2.1 JOIN	Uses the SQL JOIN clause to merge data.
		2.2.2 Merge Columns	Merges two columns from two tables.
		2.2.3 Merge Rows (UNION)	Uses the SQL UNION operator to merge rows.
		2.3.1 Standardization	Standardizes a column in a table.
		2.3.2 Split	Splits data based on the specified ratio.

Level 1	Level 2	Level 3	Description
	2.3 Others	2.3.3 Normalization	Normalization is a method to simplify computation. Normalization converts a dimensional expression into a dimensionless expression, which is a scalar.
		2.3.4 Missing Data Imputation	Replaces a null or specified value with the maximum, minimum, average, or custom value.
		2.3.5 KV to Table	Converts key-value (KV) pairs to a standard table.
		2.3.6 Table to KV	Converts a standard table to KV pairs.
		2.3.7 Add ID Column	Adds an auto-increment ID column to a table.
		2.3.8 Data Type Conversion	Converts data types.
	3.1 Feature Transformation	3.1.1 Principal Component Analysis (PCA)	A dimensionality reduction algorithm.
		3.1.2 Singular-value Decomposition (SVD)	An algorithm that is used to decompose matrices in linear algebra. It is a generalization of the diagonalization of normal matrices in matrix analysis.
		3.1.3 Feature Scaling	A computer vision algorithm that is used to detect and describe local features in images. It finds extreme points in a space scale and extracts their positions, scales, and rotation invariants.
		3.1.4 Feature Anomaly Smoothing	Scales anomalies in the input features to a specified interval. This component supports both sparse and dense data formats.

Level 1	Level 2	Level 3	Description
3.Feature Engineering		3.1.5 Feature Discretization	Supports discretization of dense numeric features, and unsupervised discretization such as equal frequency discretization and equal width discretization. It also supports supervised discretization such as Gini gain-based discretization and entropy gain-based discretization.
		3.1.6 Anomaly Detection	Detects anomalous features in continuous value features based on the maximum value and minimum value of the box plot. It can also filter out anomalous features in enumerated value features by using the threshold that is specified based on how frequently the feature values are enumerated.
	3.2 Feature Importance	3.2.1 Linear Model Feature Importance	Evaluates the importance of the features in the corresponding model.
		3.2.2 Random Forest Feature Importance Evaluation	
	3.3 Feature Selection	3.3.1 Filter-based Feature Selection	Uses different feature selection methods to filter features, selects the features that rank top N in importance, and then saves all feature importance tables. This component supports both sparse and dense data formats.
	3.4 Feature Generation	3.4.1 One Hot Encoding	Provides training and prediction features. You can use this component to encode data in one-hot mode or use the existing one-hot model to encode new data in one-hot mode.
		3.4.2 Feature Encoding	Uses decision trees and ensemble methods to explore new features. The features are the one-hot encoding results of the leaf nodes of decision trees. These nodes are composed of one or more features.
	4.1 Percentile	-	Calculates the percentile of a column.

Level 1	Level 2	Level 3	Description
4.Statistics	4.2 Data Pivoting	-	Allows you to perform data pivoting.
	4.3 Covariance	-	Measures the covariance of two specified values.
	4.4 Empirical Probability Density Chart	-	Returns nonparametric distribution based on the estimated probability distribution.
	4.5 Chi-square Goodness of Fit Test	-	Determines the differences between the observed and expected frequencies of each class for a multiclass nominal variable.
	4.6 Chi-square Test of Independence	-	Determines whether two factors of which each has at least two classes are mutually independent. The null hypothesis is that the two factors are independent of each other.
	4.7 Two Sample T Test	-	A two-sample t-test involve independent samples (unpaired samples) or paired samples. The two samples that are independent of each other are called independent samples. An independent samples t-test checks whether two samples significantly differ from each other. A paired sample t-test checks whether the mean values from two paired populations significantly differ from each other.
	4.8 One Sample T Test	-	Determines whether a significant difference exists between the overall mean value of a variable and a specific value.
	4.9 Normality Test	-	Determines whether observed values are normally distributed.
	4.10 Lorenz Curve	-	Illustrates the distribution of wealth across a population.
	4.11 Whole Table Statistics	-	Calculates the statistical information of each column in a table, including the default value, maximum value, minimum value, variance, and deviation.

Level 1	Level 2	Level 3	Description
	4.12 Pearson Coefficient	-	Calculates the Pearson coefficient of two numerical columns.
	4.13 Histogram	-	Displays multiple metrics in a histogram.
	4.14 Scatter Plot	-	Generates a scatter plot that shows the distribution of data points in a Cartesian coordinate system.
	4.15 Correlation Coefficient Matrix	-	Calculates a matrix of correlated coefficients for multiple columns.
	4.16 Box Plot	-	Displays box plots and jitters between multiple continuous variables and a numeration feature in a chart.
	5.1 Binary Classification	5.1.1 Linear SVM	A supervised machine learning algorithm that is used to identify and classify models, and then perform regression analysis.
		5.1.2 Logistic Regression for Binary Classification	A supervised machine learning algorithm that uses logistic regression to train a binary classification model.
		5.1.3 GBDT Binary Classification	An iterative decision tree algorithm that calculates results based on the final conclusions of multiple decision trees. GBDT is short for Gradient Boosting Decision Tree.
		5.1.4 PS-SMART Binary Classification Training	A parameter server (PS) is used to process a large number of offline and online training tasks. SMART is short for scalable multiple additive regression tree. PS-SMART is an iteration algorithm that is implemented by using a PS-based GBDT. This component supports training tasks for tens of billions of samples and hundreds of thousands of features. It can run training tasks on thousands of nodes. This component also supports multiple data formats and optimization technologies such as approximation by using histograms.

Level 1	Level 2	Level 3	Description
5.Modeling	5.2 Multiclass Classification	5.2.1 Logistic Regression for Multiclass Classification	A linear regression algorithm that is used for multiclass classification.
		5.2.2 Random Forest	A type of classifier that uses multiple trees and sample data to generate models and make predictions.
		5.2.3 KNN	For a row in a prediction table, this component selects K nearest records from the training table. Then, it adds the row to the class that is most common among the K records.
		5.2.4 Naive Bayes	A family of classification algorithms based on the theorem of Bayes and independent hypothesis of feature conditions.
		5.2.5 PS-SMART Multiclass Classification	A PS is used to process a large number of offline and online training tasks. PS-SMART is an iteration algorithm that is implemented by using a PS-based GBDT. This component supports training tasks for tens of billions of samples and hundreds of thousands of features. It can run training tasks on thousands of nodes. This component also supports multiple data formats and optimization technologies such as approximation by using histograms.
	5.3 Regression	5.3.1 GBDT Regression	An algorithm that uses the GBDT structure for regression.
		5.3.2 PS Linear Regression	An algorithm that supports a large amount of training data by using PSs.
		5.3.3 Linear Regression	Analyzes the linear relationship between a dependent variable and multiple independent variables.

Level 1	Level 2	Level 3	Description
		5.3.4 PS-SMART Regression	A PS is used to process a large number of offline and online training tasks. PS-SMART is an iteration algorithm that is implemented by using a PS-based GBDT. This component supports training tasks for tens of billions of samples and hundreds of thousands of features. It can run training tasks on thousands of nodes. This component also supports multiple data formats and optimization technologies such as approximation by using histograms.
		5.4.1 K-means Clustering	Clustering similarity is calculated based on a central object that is obtained by using the mean values of objects in different clusters.
		5.4.2 DBSCAN	Density-based spatial clustering of applications with noise (DBSCAN) is an unsupervised clustering algorithm based on data density. In a cluster, DBSCAN uses specified radius and quantity thresholds to filter core points and neighbors in a region. DBSCAN uses density-reachable and density-connected theories to cluster data points.
	5.4 Clustering		
	5.5 Evaluation	5.5.1 Binary Classification Evaluation	Evaluates and predicts the preceding algorithms.
		5.5.2 Multiclass Classification Evaluation	
		5.5.3 Regression Model Evaluation	
		5.5.4 Clustering Model Evaluation	
		5.5.5 Confusion Matrix	
	5.6 Prediction	5.6.1 Prediction	

Level 1	Level 2	Level 3	Description
	5.7 Recommendation	5.7.1 Collaborative Filtering (etrec)	A collaborative filtering algorithm that is based on items. It uses two input columns and provides the top K items with the highest similarity as the output.
6.Time Series	6.1 x13_arima	-	ARIMA is short for Autoregressive Integrated Moving Average Model. x13-arima is an ARIMA algorithm for seasonal adjustment based on the open source X-13ARIMA-SEATS.
	6.2 x13_auto_arima	-	Automatically selects an ARIMA model based on the Gomez and Maravall processes.
	7.1 Word Splitting	-	An algorithm that is used to split words in the specified text. Only Chinese is supported for the Taobao and Internet word splitting models.
	7.2 Word Frequency Statistics	-	On the basis of the word splitting results, this component generates the words in their original order and calculates the number of times each word appears in the documents that is specified by the document ID column.
	7.3 TF-IDF	-	A statistical method that is used to evaluate the importance of a word within a document in a collection or corpus.
	7.4 PLDA	-	Provides probability distribution of the topics of each document.
	7.5 Word2Vec	-	Converts words to vectors.
	7.6 Convert Row, Column, and Value to KV Pair	-	Converts a set of a row, a column, and a value to a KV pair (row, [col_id,value]).
	7.7 Text Summarization	-	Uses algorithms to create a summary on a document.
	7.8 Keyword Extraction	-	Extracts words from a document. The extracted words are most correlated to the meaning of the document.
	7.9 Sentence Splitting	-	Splits sentences by punctuation.

Level 1	Level 2	Level 3	Description
7.Text Analysis	7.10 Deprecated Word Filter	-	A preprocessing method that is used for text analysis. This method is used to filter out the noise in word splitting results, such as of, yes, and ah. Custom dictionaries are supported.
	7.11 String Similarity	-	String similarity calculation is a basic operation in machine learning. It is typically used in scenarios such as information retrieval, natural language processing, and bioinformatics. This algorithm supports the following similarity calculation methods: Levenshtein distance, longest common substring, string subsequence kernel, cosine, and simhash_hamming. It supports two input methods: string-to-string calculation and top N calculation.
	7.12 String Similarity - Top N	-	Checks whether a given string is one of the top N predictions.
	7.13 Semantic Vector Distance	-	Generates words or sentences that are nearest to a given word or sentence based on the semantic vectors that are calculated by an algorithm component such as Word2vec. For example, you can use this component to generate a list of words that are most similar to a given word based on the semantic vectors that are returned by the Word2vec component.
	7.14 N-gram Counting	-	N-grams are generated based on words. The number of the N-grams in all corpora is counted.
	7.15 Pointwise Mutual Information (PMI)	-	Counts the co-occurrence of all words in several documents and calculates the PMI between every two words.
	7.16 Document Similarity	-	Calculates the similarity between documents or sentences that are separated with spaces based on the similarity of strings. The document similarity is calculated in the same way as string similarity calculation.

Level 1	Level 2	Level 3	Description
	7.17 Doc2Vec	-	Maps text to vectors.
	7.18 Conditional Random Field	-	A conditional probability distribution model that generates another set of random variables based on a set of input random variables. This model presumes that the output random variables constitute a Markov random field (MRF). Conditional random fields can be used for prediction in different labeling scenarios, such as a linear chain.
	7.19 Conditional Random Field Prediction	-	
8.Deep Learning	8.1 TensorFlow 1.8	-	Uses the PAI-TensorFlow framework to implement deep learning.
	8.2 PyTorch	-	Uses the PAI-PyTorch framework to implement deep learning.
	9.1 K-Core	-	The k-core of a graph is the subgraph that remains after all vertices with a degree less than or equal to K are removed.
	9.2 Single-source Shortest Path	-	Calculates the shortest path between two points.
	9.3 Page Rank	-	Calculates the rank of a web page.
	9.4 Label Propagation Clustering	-	A semi-supervised machine learning algorithm based on graphs. A node is labeled based on the labels of its neighboring nodes. The scale at which labels are propagated is determined by the similarity between the node and its neighbors. Data becomes stable by iterative propagation updates.
	9.5 Label Propagation Classification	-	A semi-supervised classification algorithm that uses the label information of labeled nodes to predict that of unlabeled nodes.

9.Network Level 1 Analysis	Level 2	Level 3	Description
	9.6 Modularity	-	Modularity is a metric that is used to evaluate the structure of communities in a network. It is designed to measure the closeness of communities divided from a network structure. A value greater than 0.3 indicates a strong community structure.
	9.7 Maximal Connected Subgraph	-	A maximum connected subgraph of an undirected graph G is a connected subgraph of G, where all vertices are connected with the least edges.
	9.8 Vertex Clustering Coefficient	-	Calculates the peripheral density of nodes near a node in an undirected graph G. The density of a star network is 0. The density of a fully meshed network is 1.
	9.9 Edge Clustering Coefficient	-	Calculates the peripheral density of each edge in an undirected graph.
	9.10 Counting Triangle	-	Generates all triangles in an undirected graph.
	9.11 Tree Depth	-	Generates the depth and tree ID of each node in a network that contains multiple trees.
10.Tools	10.1 MaxCompute SQL	-	Runs MaxCompute SQL statements.
	11.1 Binning	-	Performs data binning in equal frequency, equal width, or auto binning mode. This component uses continuous or discrete features as the input, and generates binning rules for all features.
	11.2 Data Conversion Module	-	Uses binning results to convert the input features that you provide. You can use this component to perform normalization, Weight of Evidence (WoE) conversion, and discretization. Normalization generates values between 0 to 1. WoE replaces feature values with WoE of bins. Discretization converts variables to dummy variables based on binning results. All output data is in the KV format.

Level 1	Level 2	Level 3	Description
11.Financial Section	11.3 Scorecard Training	-	The scorecard is a modeling tool that is commonly used in credit risk evaluation. Scorecard modeling performs discretization on original variables by using binning. Then, it uses linear models, such as logistic regression and linear regression, to conduct model training. The scorecard supports feature selection and score conversion. In addition, it allows you to add constraints to variables during model training.
	11.4 Scorecard Prediction	-	Uses the model that is generated by the Scorecard Training component to predict scores.
	11.5 Population Stability Index	-	Population stability index (PSI) offers you a measurement standard to identify a shift in the population over a period of time. It indicates the stability of the population based on changes, for example, the changes within two months. A PSI value less than 0.1 indicates insignificant changes. A PSI value between 0.1 and 0.25 indicates changes that are relatively significant. A PSI value greater than 0.25 indicates sharp changes and requires special attention.
	12.1 Video Processing	12.1.1 Data to TFRecord	Converts labeled data to TFRecord files, which can be used in subsequent training.
		12.1.2 General Video Processing	Extracts video frames. You can customize the methods for frame extraction and specify the intervals at which extraction is performed based on your business requirements.
		12.2.1 Image Segmentation Training	Implements semantic segmentation based on DeepLab V3.
		12.2.2 Image Detection Training	Trains image detection models.

Level 1	Level 2	Level 3	Description
12. Video-based Intelligent Algorithm	12.2 Offline Training Model	12.2.3 Image Classification Training	Trains image classification models.
		12.2.4 Video Classification Training	Supports regular 3D convolutional neural network (CNN) models. This component can be used to facilitate video classification training.
		12.2.5 End-to-End Text Recognition Training	Trains end-to-end optical character recognition (OCR) models.
	12.3 Offline Prediction Model	12.3.1 General Image Prediction	Makes predictions in image classification, detection, and segmentation.
		12.3.2 General Video Prediction	Makes predictions in video classification. You can use this component to predict the results of video classification and feature extraction tasks.
		12.3.3 ASR Prediction	An automatic speech recognition (ASR) algorithm that can be used to make predictions in speech recognition.

 **Note** Deep learning components and video-based intelligent algorithm components are applicable to Intel servers, but not Hygon servers.

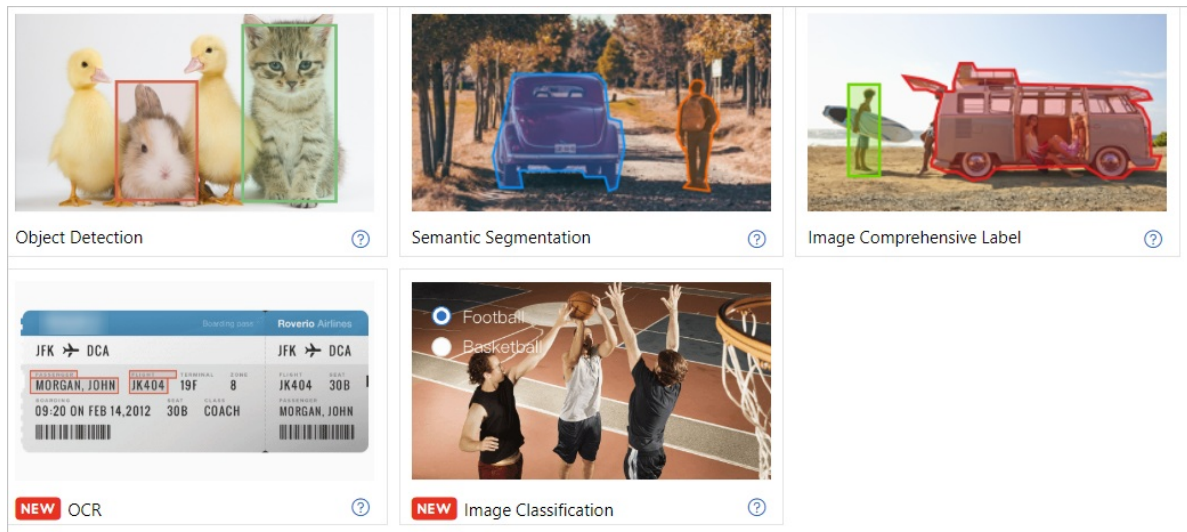
27.4.7. Data labeling

This topic describes the data labeling feature of Machine Learning Platform for AI (PAI).

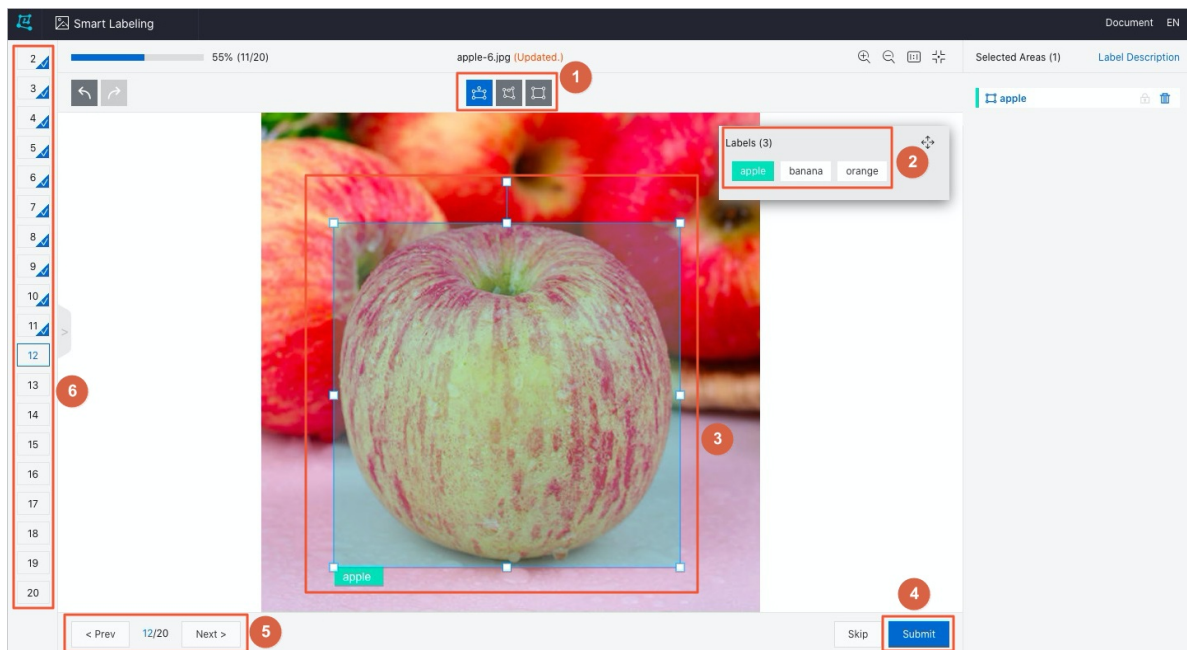
 **Note** You can use the data labeling feature only on Intel servers instead of Hygon servers.

AI integrates data, algorithms, and business scenarios. In actual implementation of AI, data quality greatly affects the accuracy of algorithm models. In view of this, PAI provides the data labeling feature. You can use this feature to efficiently label data. This reduces costs in data preparation. The data labeling feature has the following benefits:

- Provides multiple built-in labeling templates that are ready for you to use in regular labeling scenarios.



- Allows you to flexibly distribute labeled topics to different users for collaborative labeling.
- Provides an easy-to-use interface and supports shortcut keys for regular operations. This improves data labeling efficiency.



27.4.8. DSW

This topic describes the basic features of Data Science Workshop (DSW) of Machine Learning Platform for AI (PAI). This topic also shows you the interfaces related to modeling in DSW.

Features

PAI supports visualized modeling and allows you to create models in Notebook or an online integrated development environment (IDE). This way, native code can be used to create custom models to better meet your requirements. You can select a programming environment as needed on the Notebook Modeling page to create and train models. DSW provides the following features:

- Allows you to start and stop DSW instances on demand, save images with a few clicks, restore development environments, and access resources over virtual private clouds (VPCs).

- Provides integrated AI development environments.
- Provides built-in big data development packages and algorithm packages, and grants sudo permissions to you for installing third-party libraries.
- Provides official images that support different versions of mainstream computing frameworks, such as TensorFlow and PyTorch.
- Provides built-in WebIDE that allows you to install all plug-ins.

Interfaces related to modeling in DSW

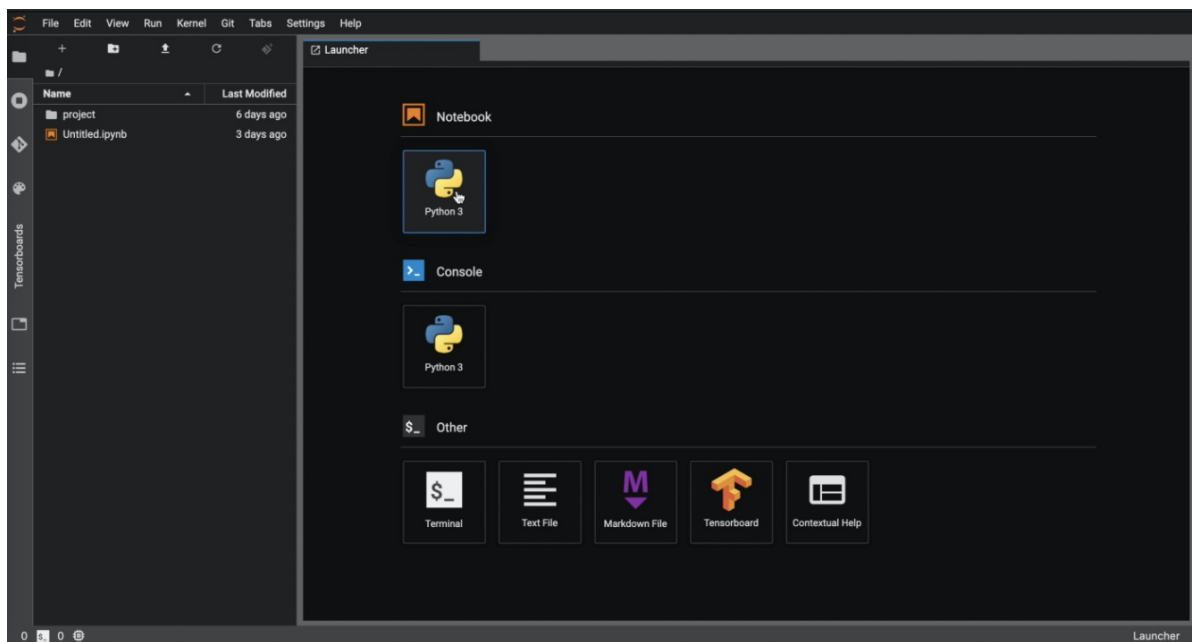
Modeling in DSW involves the following interfaces:

- DSW-Notebook Service page in the PAI console

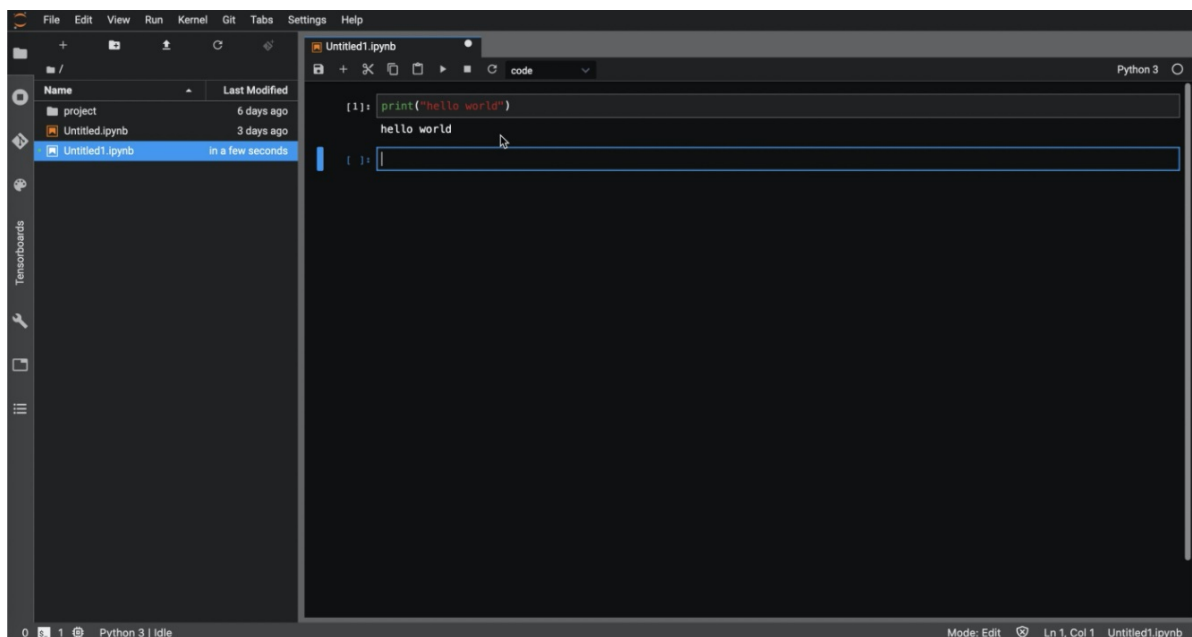
In the PAI console, you can view the information about all created DSW instances and their statuses. You can create, stop, or delete DSW instances. In addition, you can click Lab in the Actions column to create models in Notebook. Alternatively, you can click IDE or Terminal in the Actions column to create models in WebIDE based on native code, or create models on online terminals.

- Notebook modeling interface

You can create, upload, or open an existing notebook in the Notebook modeling interface, as shown in the following figure. Notebook supports various data formats. This facilitates data exploration and modeling.

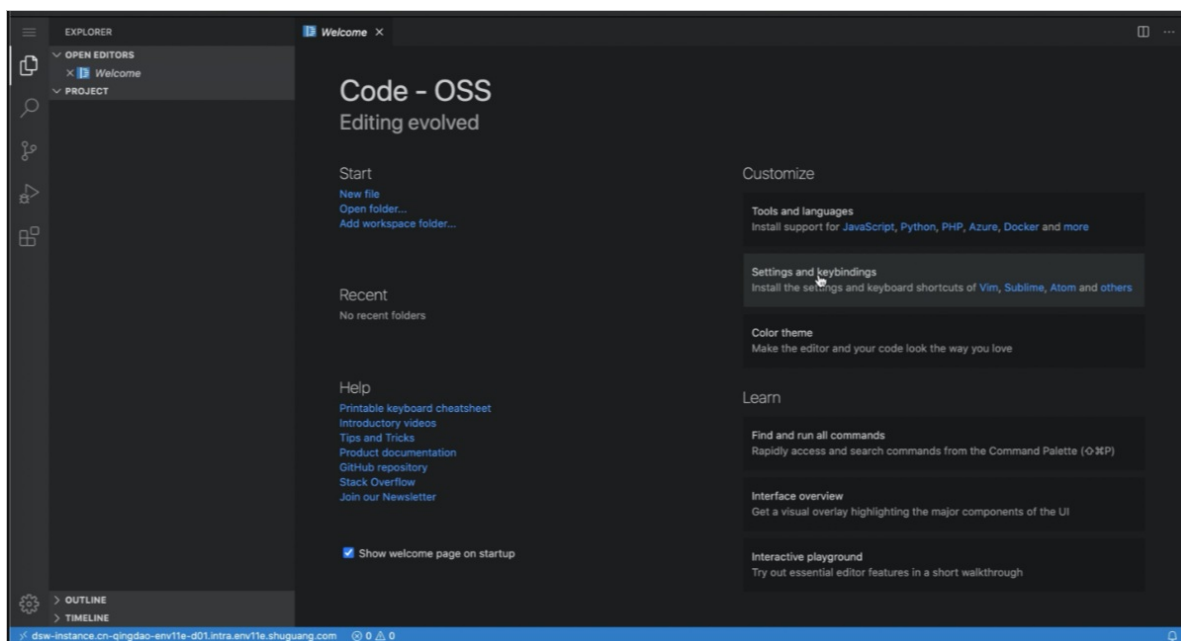


The following figure shows how to create a model in Notebook.



- WebIDE interface

You can use native code to create models in WebIDE.



- Online terminal interface

You can use command lines to create models on online terminals.

```
~/workspace> mount | grep workspace
13478489e7-qxn69.cn-qingdao-env11e-d01.nas.intra.env11e.shuguang.com:/ on /home/admin/workspace type nfs4 (rw,relatime,vers=4.0,rsize=1048576,wsiz=1048576,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=10.34.1.60,local_lock=none,addr=10.11.80.187)
~/workspace>
```

Persistent storage

If the DSW environment contains a distributed storage system such as a bucket in Object Storage Service (OSS) or a file system in Apsara File Storage NAS (NAS), PAI automatically creates a disk for each logon user in OSS or NAS. The disk is a remote disk that is bound to each logon user and mounted to the fixed directory `/home/admin/workspace`. After a DSW instance is created, DSW mounts the created disk to the fixed directory `/home/admin/workspace`.

By default, all operations related to modeling in Notebook or WebIDE or on terminals are performed in this directory. The operation data that is generated when you write code, obtain data files, and export training results in this directory is permanently stored on the remote disk.

The remote disk separates computing from storage, and provides the following benefits:

- If your log off from your DSW instance for specific reason, user data is not lost.
- When you start your DSW instance for computing, the disk is automatically mounted to your instance, and you are connected to the data stored on the disk.

27.5. Scenarios

Machine Learning Platform for AI can be applied to the following scenarios:

Marketing

- Use cases: commodity recommendations, user profiling, and precise advertising.
- Example: Machine Learning Platform for AI associates user shopping behavior data with commodities to offer commodity recommendations and evaluate the recommendation results.

Finance

- Use cases: loan delivery prediction, financial risk management, stock trend prediction, and gold price prediction.
- Example 1: Agricultural loan delivery is used in a typical example of data mining. A lender uses machine learning to build an empirical model based on historical data such as the lendee's annual income, cultivated crop type, and debit and credit history. This model is then used to predict the lendee's capacity.
- Example 2: Users' credit card expense records are processed by a machine learning algorithm. After raw data binning and feature engineering transformation, data is used to build a linear model. The final credit score of each user is determined by the model predictions, and can be used in a variety of loan and finance related credit checks.

Text

- Use cases: news classification, keyword extraction, text summarization, and text analysis.
- Example: A simple system for automatic commodity label classification is built using the text analysis function of Machine Learning Platform for AI.

Take online shopping as an example. A commodity typically has labels for multiple dimensions. For example, the commodity description of a pair of shoes may be "Korean Girl Dr. Martens Women's Preppy/British-style Lace-up Dull-polish Ankle High Platform Leather Boots." A bag may be described as "Discount Every Day 2016 Autumn and Winter New Arrival Women's Korean-style Seashell-shaped Tassel Three-way Bag as a Messenger Bag, Hand Carry Bag, and Shoulder Bag."

Each product description contains multiple dimensions such as the time, place of origin, and style. E-commerce platforms face the daunting challenge of how to classify hundreds of thousands of products based on these specified dimensions. The biggest challenge is determining which labels constitute the dimensions of each product. A label classification system can be built using a machine learning algorithm to automatically learn label terms. For example, the system can learn location-related labels such as Japan, Fujian, and Korea.

Unstructured data processing

- Use cases: image classification and image text extraction by using optical character recognition (OCR).
- Example: A prediction model can be quickly built for image recognition by using the TensorFlow deep learning framework. The TensorFlow deep learning framework can recognize images and return image classification results within half an hour. Image recognition by using deep learning can also be used in illicit image filtering, facial recognition, and object detection.

Other prediction cases

- Use cases: rainfall forecast, football match result prediction, microblog leader analysis, and social relationship chain analysis.
- Example: Air quality can be predicted by Machine Learning Platform for AI based on historical air quality index data such as PM 2.5, carbon monoxide concentration, and nitrogen dioxide concentration. The prediction results can then be used to determine which air quality index has the greatest impact on PM 2.5 levels.

27.6. Limits

This topic describes the limits of Machine Learning Platform for AI.

Item	Description
MaxCompute service deployment	The computing service of Machine Learning Platform for AI relies on MaxCompute to store tables and perform some SQL-related computations. Therefore, MaxCompute must be deployed before you can use the machine learning service.
OSS service deployment (deep learning)	The deep learning service of Machine Learning Platform for AI relies on OSS to store, read, and write data. Therefore, OSS must be deployed.
Limit to component use	For more information, see parameter configurations for each component.

27.7. Terms

This topic describes terms used in Machine Learning Platform for AI.

data mining

A broad definition that describes the use of algorithms to explore useful information from large amounts of data. Typically, data mining uses machine learning algorithms.

Alibaba Cloud DTplus

The big data platform of Alibaba Cloud. DTplus provides enterprises with a complete set of end-to-end big data solutions for fields such as enterprise data warehouses, BI, machine learning, and data visualization. These solutions help enterprises become more agile, smarter, and more perceptive in the data technology (DT) era.

table

Data storage units of MaxCompute. Tables used by machine learning are stored in MaxCompute. Logically, a table is a two-dimensional structure that consists of rows and columns. Each row represents a record. Each column represents a field of the same data type. One record can contain one or more columns. The schema of a table consists of column names and column types.

On Machine Learning Platform for AI, you can create a table, add the table to favorites, and import data to the table. The table is automatically stored in MaxCompute. To delete a table, you must log on to MaxCompute.

partition

Certain columns specified in a table when the table is created. In most cases, you can consider a partition as a directory in a file system.

Tables are stored in MaxCompute. MaxCompute uses each value in a partition column as a partition (directory). You can specify multiple hierarchies of partitions by using multiple table columns as table partitions. The relationships between partitions are similar to those between multiple hierarchies of directories.

When using data, if you specify the name of a partition, only the data in the specified partition is read. This removes the need to scan the entire table for data, improves processing efficiency, and minimizes costs.

lifecycle

The period of time that determines how long a table partition is retained since it was last updated. If a table (partition) is not updated within the specified time period, MaxCompute automatically deletes it.

sparse data format

Datasets in which most data entries are null or have a value of 0. Sparse data can be utilized effectively if efficient methods are used to explore the useful information that exists in the relatively incomplete sparse data set.

On Machine Learning Platform for AI, if the data of a feature in a sample is in the sparse format, you must convert the format to the LibSVM format, select **key:value**, **key:valuesparse data format** on the parameter setting page, and then upload the data.

feature

An attribute that is used to describe an object. For example, a person can be described by age, gender, occupation, and other attributes. Each of these attributes is a feature of the person.

On Machine Learning Platform for AI, a dataset is stored as a table. A column in the table is a feature of this dataset. The features of data are important to machine learning. Data and its features determine the upper limit of machine learning capabilities. Models and algorithms are used to help machine learning reach the upper limit. Therefore, features must be processed before a machine learning experiment can be executed. Typical feature processing methods include data preprocessing, feature selection, and dimension reduction.

dimension reduction

A method that is used to remove the dimensions that have minor impacts and extract key features from a large number of features. A dimension is the way something is observed. In machine learning, dimensions describe the features of a dataset. If a dataset has millions of features, the training model for machine learning will be complex and the training will take a long period of time. In this case, dimension reduction is required. The dimension reduction algorithms on Machine Learning Platform for AI include PCA and LDA.

28.DataHub

28.1. What is DataHub?

DataHub is a platform designed to process streaming data. You can publish and subscribe to streaming data in DataHub and distribute the data to other platforms. DataHub allows you to analyze streaming data and build applications based on the streaming data.

DataHub collects, stores, and processes streaming data from mobile devices, applications, website services, and sensors. You can use your own applications or Apsara Stack Realtime Compute to process streaming data in DataHub, such as real-time website access logs, application logs, and events. The processing results such as alerts and statistics presented in graphs and tables are updated in real time.

Based on the Apsara system of Alibaba Cloud, DataHub features high availability, low latency, high scalability, and high throughput. DataHub is seamlessly integrated with Realtime Compute, allowing you to use SQL to analyze streaming data.

DataHub can also distribute streaming data to Apsara Stack services such as MaxCompute and Object Storage Service (OSS).

28.2. Benefits

High throughput

You can write terabytes (TB) of data into a topic and up to 80 million records into a shard every day.

Real-time processing

DataHub makes it easy to collect and process various types of streaming data in real time so you can react quickly to new information.

Ease of use

- DataHub provides a variety of SDKs for C++, Java, Python, Ruby, and Go.
- In addition to SDKs, DataHub provides RESTful APIs so that you can manage DataHub by using existing protocols.
- You can use collection tools such as Fluentd, Logstash, and Oracle GoldenGate to write streaming data into DataHub.
- DataHub supports structured and unstructured data. You can write unstructured data to DataHub, or create a schema for the data before it is written into the system.

High availability

- The processing capacity of DataHub is automatically scaled out without affecting your services.
- DataHub automatically stores multiple copies of data.

Scalability

You can dynamically adjust the throughput of each topic. The maximum throughput of a topic is 256,000 records per second.

Data security

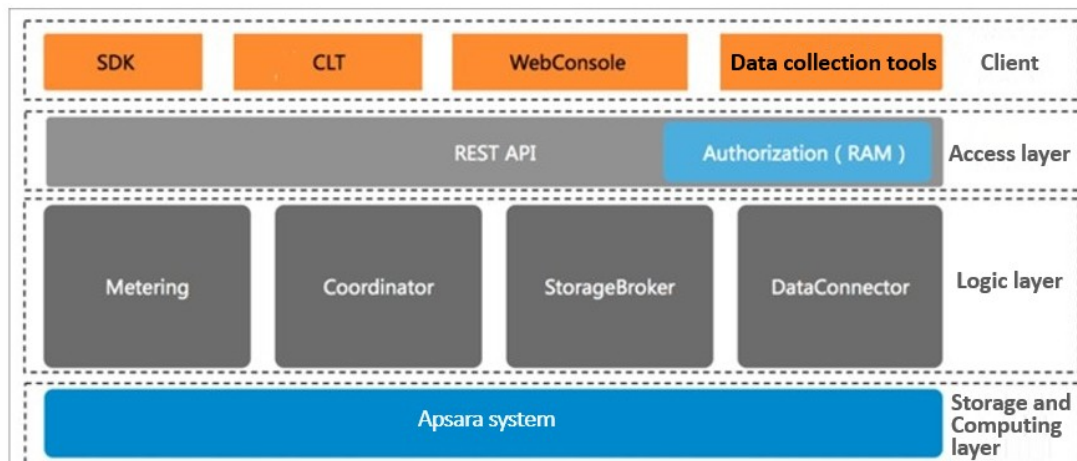
- DataHub provides enterprise-level security measures and isolates resources between users.

- It also provides several authentication and authorization methods, including whitelist configuration and RAM user management.

28.3. Architecture

Architecture shows the architecture of DataHub.

Architecture



The architecture of DataHub consists of four layers: **clients**, **access layer**, **logic layer**, and **storage and scheduling layer**.

Clients

DataHub supports the following types of clients:

- **SDKs:** DataHub provides SDKs in a variety of languages such as C++, Java, Python, Ruby, and Go.
- **Command-line tools (CLTs):** You can run commands in Windows, Linux, or Mac operating systems to manage projects and topics.
- **Console:** In the console, you can manage projects and topics, create subscriptions, view the shard status, monitor topic performance, and manage DataConnectors.
- **Data collection tools:** You can use Logstash, Fluentd, and Oracle GoldenGate (OGG) to collect data to DataHub.

Access layer

You can access DataHub by using HTTP and HTTPS. DataHub supports Resource Access Management (RAM) authorization and horizontal scaling of topic performance.

Logic layer

The logic layer handles the key features of DataHub, including project and topic management, data read and write, offset-based data consumption, traffic statistics, and data synchronization. Based on these key features, the logic layer is composed of the following modules: StorageBroker, Metering, Coordinator, and DataConnector.

- **StorageBroker:** provides data reads and writes in DataHub. This module adopts the log file storage model of Apsara Distributed File System, halving the read/write volume compared with the conventional write-ahead logging (WAL) model. This module stores three copies of data to ensure that no data is lost if a server fault occurs, and supports disaster recovery between data centers. It supports real-time data caching to ensure efficient consumption of real-time data and supports an

independent read cache of historical data to enable concurrent consumption of historical data.

- Metering: supports shard-level billing based on the consumption period.
- Coordinator: supports offset-based data consumption and horizontal scaling of the processing capacity. It supports up to 150,000 QPS on a single node.
- DataConnector: supports automatic data synchronization from DataHub to other Apsara Stack services, including MaxCompute, OSS, AnalyticDB, ApsaraDB RDS for MySQL, Tablestore, and Elasticsearch.

Storage and scheduling layer

- Storage: Based on the log file storage model of Apsara Distributed File System, DataHub supports append operations and solid state drive (SSD) storage. Data in each shard is stored in a separate file based on the timestamp of the data.
- Scheduling: Based on Job Scheduler, DataHub assigns shards to nodes based on the traffic on each shard. This ensures that the shards do not occupy the CPU or memory of Job Scheduler. The number of partitions on a single node has no upper limit. DataHub supports failovers within milliseconds and hot upgrades.

28.4. Features

28.4.1. Data queue

DataHub automatically generates a cursor for each record in a shard. The cursor is a unique sequence of numbers. You can improve the performance of a topic by increasing the number shards in the topic.

28.4.2. Checkpoint-based data restoration

DataHub supports saving checkpoints for subscribed applications in the system. You can restore data from any checkpoint you saved if your subscribed application fails.

28.4.3. Data synchronization

Data in DataHub is automatically synchronized to other Alibaba Cloud services.

DataConnector

You can create a DataConnector to synchronize DataHub data in real time or near real time to other Alibaba Cloud services, such as MaxCompute, Object Storage Service (OSS), Elasticsearch, ApsaraDB RDS for MySQL, AnalyticDB, and Tablestore.

You can configure the DataConnector so that the data you write to DataHub can be used in other Alibaba Cloud services. At-least-once semantics is applied in data synchronization. This ensures that no data is lost, but may result in duplicated records in the destination platform if an error occurs during the synchronization process.

Destination platforms

The following table describes the platforms to which DataHub records can be synchronized.

Destination platforms

Destination platform	Timeliness	Description
MaxCompute	Near real-time. Latency: 5 minutes.	The column names and data types in the source topic must be the same as those in MaxCompute. The MaxCompute table must have one or more corresponding partition columns.
OSS	Real-time.	Records are synchronized to the specified bucket in OSS and are saved as CSV files.
Elasticsearch	Real-time.	Records are synchronized to the specified index in Elasticsearch. Records may not be synchronized in the order of the recording time. If you want to synchronize data in the order of the recording time, you must write the records with the same partition key into the same shard.
ApsaraDB RDS for MySQL	Real-time.	Records are synchronized to the specified table in ApsaraDB RDS for MySQL.
AnalyticDB	Real-time.	Records are synchronized to the specified table in AnalyticDB.
Tablestore	Real-time.	Records are synchronized to the specified table in Tablestore.

28.4.4. Scalability

The throughput of each topic can be scaled by splitting or merging shards.

You can adjust the number of shards in a topic according to the service load.

For example, if the topic throughput cannot handle a surge in the service load during Double 11, you can split existing shards to up to 256 to increase the throughput to 256 MB/s.

As the service load decreases after Double 11, you can reduce the number of shards as needed by performing the merge operation.

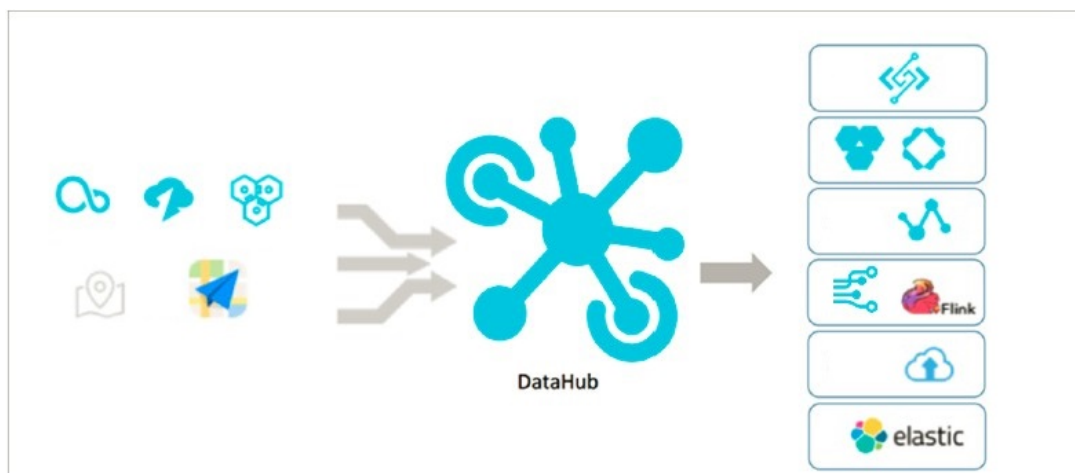
28.5. Scenarios

28.5.1. Overview

As a streaming data processing platform, DataHub can be used with various Alibaba Cloud products to provide one-stop data processing services.

28.5.2. Data uploading

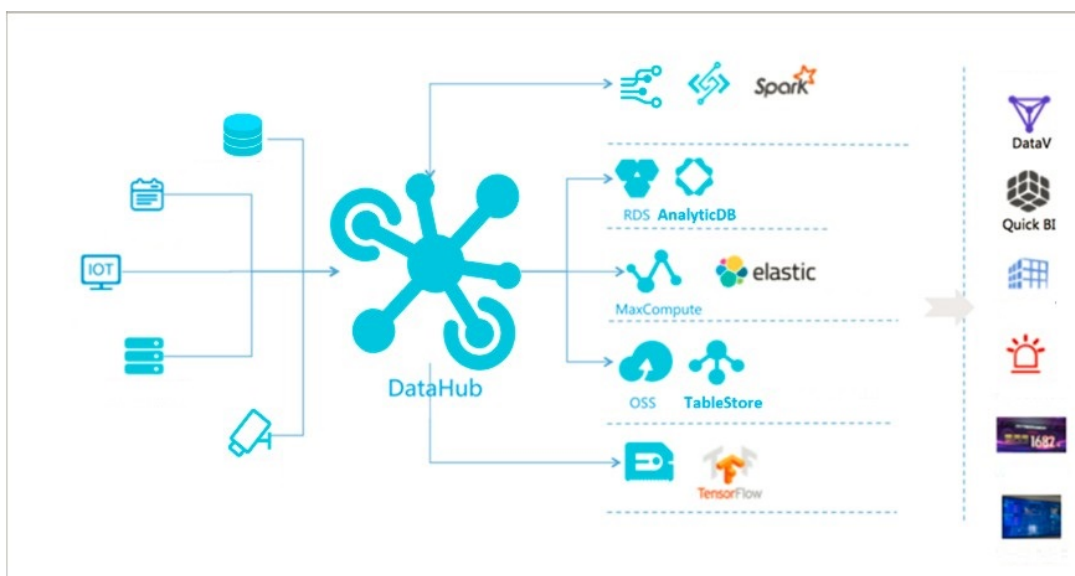
Data uploading



DataHub is connected to other Alibaba Cloud services, saving you the trouble of uploading the same data to different platforms.

28.5.3. Data collection

Data collection

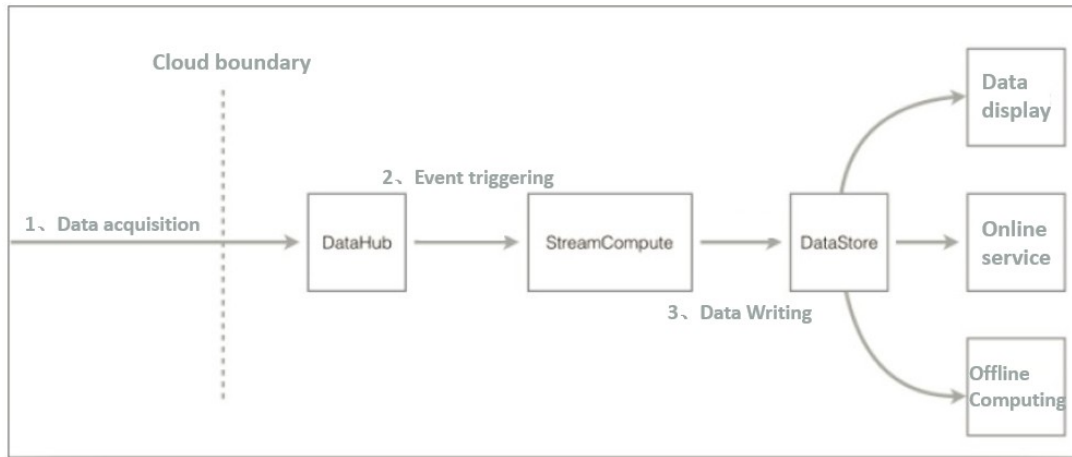


DataHub provides several types of data collection tools for you to write your data into DataHub. DataHub supports log collection from Logstash and Fluentd, and binary log collection from Data Transmission Service (DTS) and Oracle GoldenGate (OGG). DataHub also supports the collection of surveillance videos through GB28181.

28.5.4. Realtime Compute

Realtime Compute is a real-time computing engine of Alibaba Cloud, which allows you to use a language similar to SQL to analyze streaming data. Data can be transferred from DataHub to Realtime Compute or from Realtime Compute to DataHub.

DataHub and Realtime Compute



28.5.5. Data utilization

You can build an application to consume the data in DataHub, process the data in real time, and output the process results.

You can also use another application to process the streaming data output from the previous application to form a directed acyclic graph (DAG)-based data processing procedure.

28.5.6. Data archiving

You can create a DataConnector to periodically archive data in DataHub to MaxCompute.

28.6. Limits

Limits

Item	Limit	Description
Active shards per topic	(0,256]	Each topic can contain up to 256 active shards.
Total shards per topic	(0,512]	You can create up to 512 shards in each topic.
Http BodySize	Up to 4 MB	The size of the HTTP request body cannot exceed 4 MB.
String size	Up to 1 MB	The size of a string cannot exceed 1 MB.
Merge and split operations on new shards	5s	You cannot merge a shard with another shard or split the shard within the 5s after the shard is created.
Queries per second (QPS)	Up to 5,000	The write QPS limit for each shard is 5,000. Multiple queries in one batch are considered as one query.

Item	Limit	Description
Throughput	Up to 5 MB/s	Each shard provides a throughput of up to 5 MB/s.
Projects	Up to 100	You can create up to 100 projects with each account.
Topics per project	Up to 1,000	You can create up to 1,000 topics in each project. Contact the administrator if you need to create more topics.
Time-to-live (TTL) of records	[1,7]	The TTL of each record in a topic ranges from one to seven days.

28.7. Terms

project

A project is an organizational unit in DataHub and contains one or more topics. DataHub projects and MaxCompute projects are independent of each other. Projects that you create in MaxCompute cannot be used in DataHub.

topic

The smallest unit for data subscription and publishing. You can use topics to distinguish different types of streaming data. For more information about projects and topics, see Limits in Product Introduction.

time-to-live of records

The period that each record can be retained in the topic. Unit: day. Minimum value: 1. Maximum value: 7.

shard

A shard in a topic. Shards ensure the concurrent data transmission of a topic. Each shard has a unique ID. A shard can be in a different status. For more information about shard status, see the following table. Each active shard consumes server resources. We recommended that you create shards as needed.

Note Shard status

Status	Description
Activating	All shards in a topic are in the Activating state when the topic is created. You cannot perform read or write operations on shards because they are being activated.
Active	Read and write operations are enabled when a shard is in the Active state.
Deactivating	A shard is in the Deactivating state when it is being split or merged with another shard. You cannot perform read or write operations on the shard because it is being deactivated.
Deactivated	A shard is in the Deactivated state when the split or merge operation is completed. The shard is read-only when it is in the Deactivated state.

hash key range

The range of hash key values for a shard, which is in [Starting hash key,Ending hash key) format. The hashing mechanism ensures that all records with the same partition key are written to the same shard.

merge

The operation that merges two adjacent shards. Two shards are considered adjacent if the hash key ranges for the two shards form a contiguous set with no gaps.

split

The operation that splits one shard into two adjacent shards.

record

A unit of data that is written into DataHub.

record type

The data type of records in a topic. Tuple and blob are supported. A tuple is a sequence of immutable objects. A blob is a chunk of binary data stored as a single entity.

 Note

- The following data types are supported in a tuple topic. Tuple data types

Type	Description	Value range
Bigint	An 8-byte signed integer.  Note Do not use the minimum value (-9223372036854775808) because this is a system reserved value.	-9223372036854775807 to 9223372036854775807
String	A string. Only UTF-8 encoding is supported.	The size of a string cannot exceed 1 MB.
Boolean	One of two possible values.	Valid values: True and False, true and false, or 0 and 1.
Double	A double-precision floating-point number. It is 8 bytes in length.	-1.0×10^{308} to 1.0×10^{308}
Timestamp	A timestamp.	It is accurate to microseconds.

- In a blob topic, a chunk of binary data is stored as a record. Records written into DataHub are Base64 encoded.

29. Apsara Big Data Manager (ABM)

29.1. What is Apsara Big Data Manager?

Apsara Big Data Manager (ABM) is an operations and maintenance (O&M) platform tailored for big data services.

ABM supports the following services:

- MaxCompute
- DataWorks
- RealtimeCompute
- Quick BI
- DataHub
- Machine Learning Platform for AI

ABM supports O&M on big data services from the perspectives of business, services, clusters, and hosts. ABM also allows you to update big data services, customize alert configurations, and view the O&M history.

Onsite Apsara Stack engineers can use ABM to easily manage big data services. For example, they can view metrics, check and handle alerts, and modify configurations.

29.2. Benefits

This topic describes the benefits of Apsara Big Data Manager in the following aspects: cluster health monitoring, resource usage analysis, and graphical O&M management.

Cluster health monitoring

Allows you to monitor and configure the devices, resources, and services that are used in the clusters of big data products, and collects performance metrics in real time for dynamic display.

Resource usage analysis

Collects the runtime statuses of cluster devices, resources, and services in real time, and supports data aggregation and analysis to help you evaluate the health status of the cluster. If the evaluation result indicates potential risks in a cluster, responsible engineers would be notified immediately.

Graphical management interface

Provides a graphical user interface for performance metrics visualization and common O&M operations.

29.3. Architecture

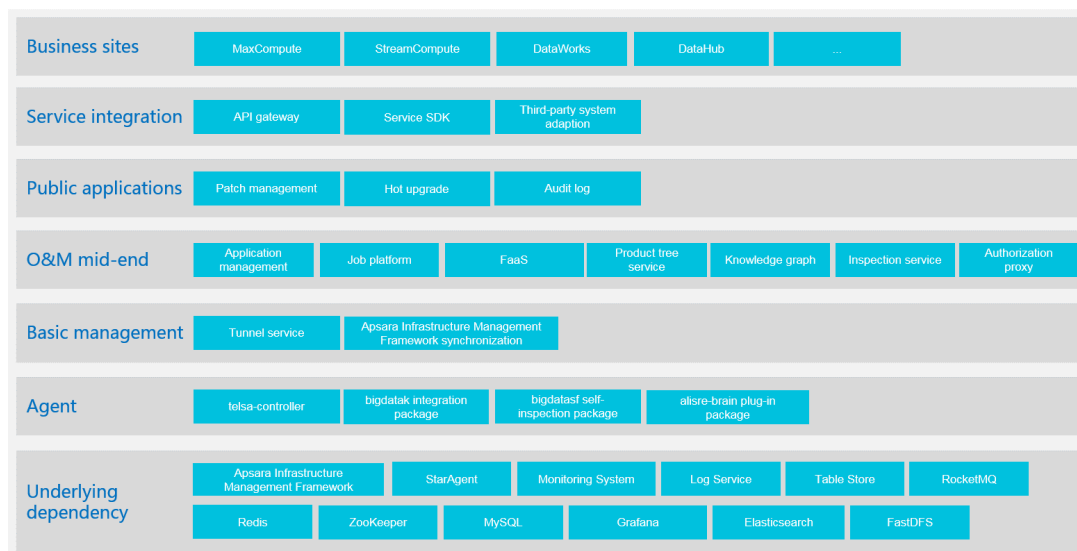
29.3.1. O&M Architecture

This topic describes the O&M architecture of Apsara Big Data Manager (ABM) and the features of each component.

ABM uses a microservice architecture that supports data integration, interface integration, and feature integration, and provides standard service interfaces. This architecture enables consistent user interfaces and O&M operations for all services in the ABM console. This reduces training costs and lowers O&M risks.

The ABM system consists of the following components: underlying dependency, agent, basic management, O&M mid-end, public applications, service integration, and business sites.

Architecture



Underlying dependency

ABM depends on Alibaba services and open source systems from third parties.

- ABM uses StarAgent and Monitoring System of Alibaba to run remote commands and execute remote data collection instructions.
- ABM uses ZooKeeper to coordinate primary and secondary services to ensure the availability of services.
- ABM uses ApsaraDB RDS to store metadata, ApsaraDB for Redis to store cache data, and Tablestore to store large amounts of self-test data. This improves service throughput.

Agent

Agent provides client SDKs, scripts, and monitoring packages that are deployed on managed servers.

O&M mid-end and basic management

The O&M mid-end and basic management components are key to ABM. Each service provides its general capabilities for business sites. This enables quick construction of business sites and makes the capabilities of each business site complete.

Public applications

Public applications are developed based on the O&M mid-end and designed with special purposes. These applications are adaptive to all big data services supported by ABM.

Service integration

Service integration links business sites with underlying components. It integrates interfaces of all internal services, adapts to various third-party systems, and provides unified SDKs for users.

Business sites

Business sites are built based on the O&M mid-end and cover all big data services such as MaxCompute, Realtime Compute for Apache Flink, DataWorks, and DataHub. Each business site provides end-to-end O&M capabilities for a service.

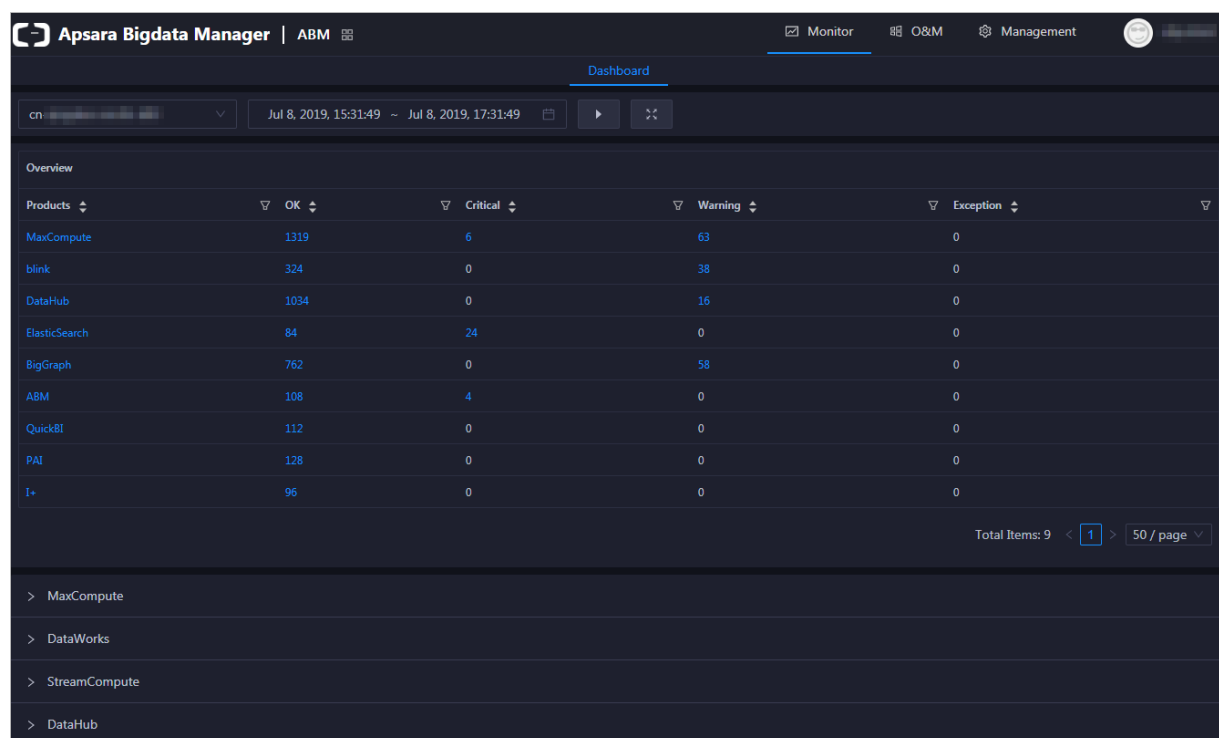
29.4. Features

29.4.1. Dashboard

Dashboard is the homepage of the Apsara Big Data Manager (ABM) console. It displays key performance metrics of MaxCompute, DataWorks, Realtime Compute for Apache Flink, and DataHub. You can also view alerts for all big data services in the dashboard. Dashboard allows you to understand the overall status of all big data services.

Dashboard page

After you log on to the ABM console, the **Dashboard** page appears by default. To return to the **Dashboard** page from another page, click the  icon in the upper-left corner and select **ABM**.



On the **Dashboard** page, you can select a region from the **Dashboard** drop-down list in the upper-left corner. You can then view the runtime performance of big data services in the selected region.

Overview of alerts

In the **Overview** section, you can view the number of alerts about each big data service. You need to pay close attention to **Critical** and **Warning** alerts, which must be cleared in a timely manner.

Products	OK	Critical	Warning	Exception
MaxCompute	1319	6	63	0
blink	324	0	38	0
DataHub	1034	0	16	0
ElasticSearch	84	24	0	0
BigGraph	762	0	58	0
ABM	108	4	0	0
QuickBI	112	0	0	0
PAI	128	0	0	0
I+	96	0	0	0

Total Items: 9 < 1 > 50 / page

In the **Overview** section, you can click a service name or the number of alerts to go to the O&M page of the service.

Overview of MaxCompute metrics

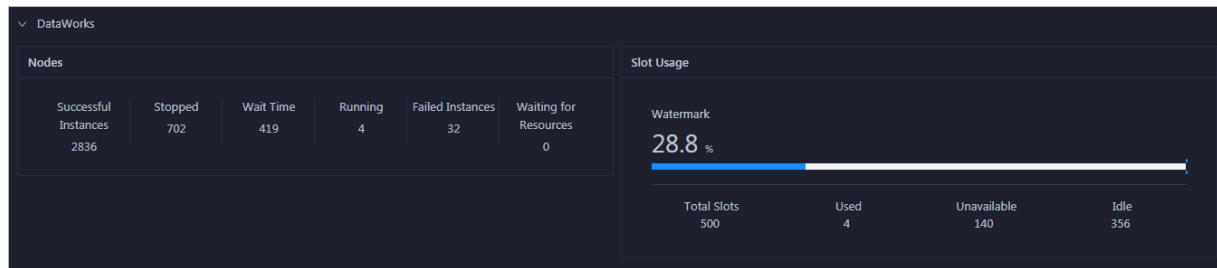
The Dashboard page displays key performance metrics of MaxCompute. To view these metrics, click **MaxCompute** in the Overview section of the Dashboard page.



The **MaxCompute** section displays the overview of jobs, real-time capacity for the control system, data traffic, compute resource usage, storage resource usage, and the trends of logical and physical CPU utilization.

Overview of DataWorks metrics

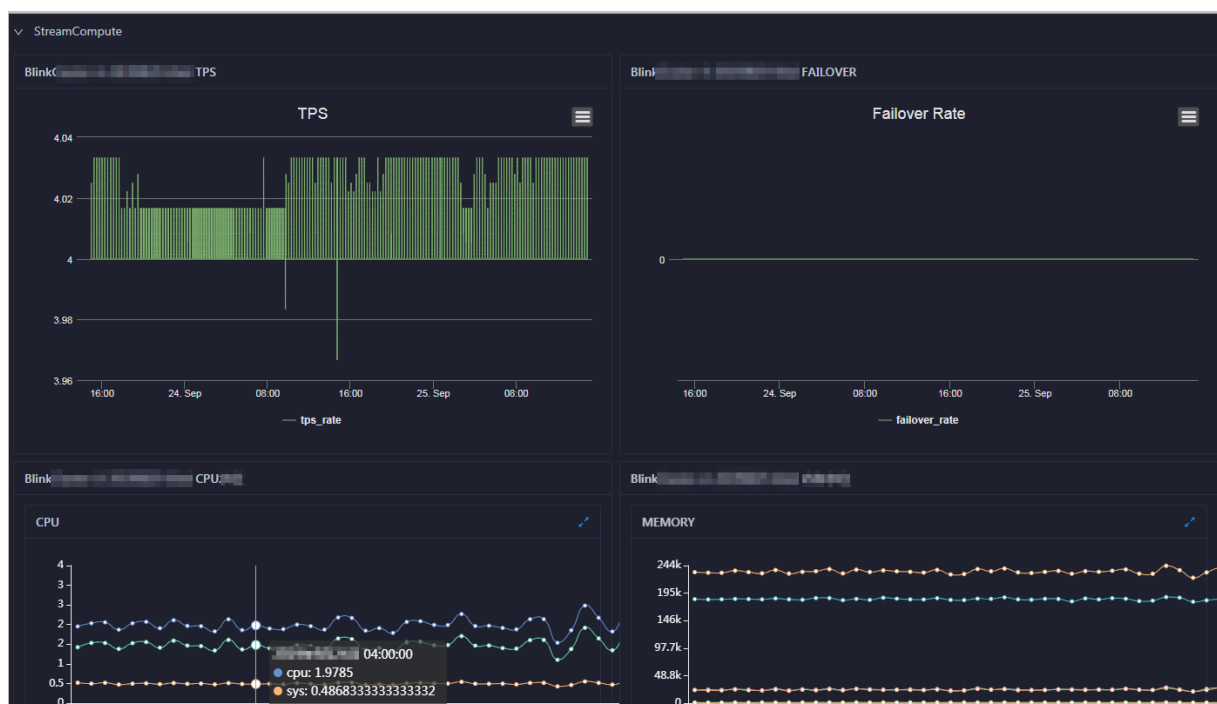
The Dashboard page displays key performance metrics of DataWorks. To view these metrics, click **DataWorks** in the Overview section of the Dashboard page.



The **DataWorks** section displays the overview of nodes, the overview of slot usage, and the trend of finished tasks.

Overview of Realtime Compute for Apache Flink metrics

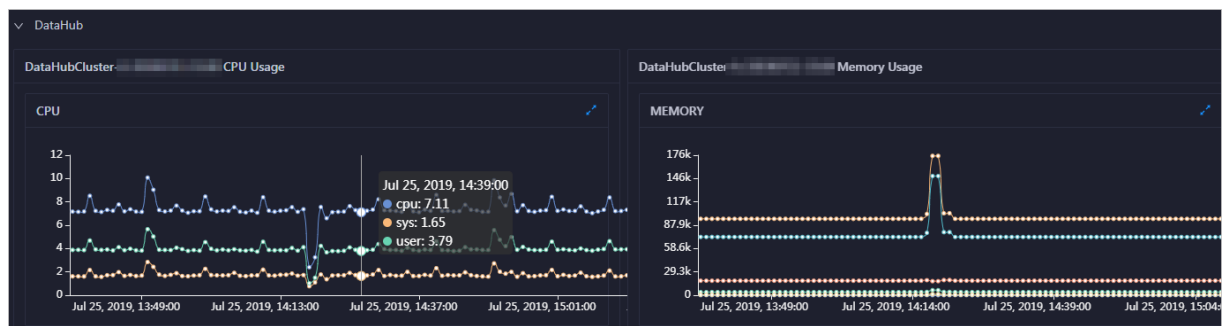
The Dashboard page displays key performance metrics of Realtime Compute for Apache Flink. To view these metrics, click **StreamCompute** in the **Overview** section of the **Dashboard** page.



In the **RealtimeCompute** section, you can view the trend charts of the transactions per second (TPS), failover rate, CPU utilization, and memory usage of a Realtime Compute for Apache Flink cluster.

Overview of DataHub metrics

The Dashboard page displays key performance metrics of DataHub. To view these metrics, click **DataHub** in the **Overview** section of the **Dashboard** page.



The **DataHub** section displays read/write latency, the numbers of read/write records, read/write request rates, read/write throughputs, and the trends of CPU utilization and memory usage.

29.4.2. Repository

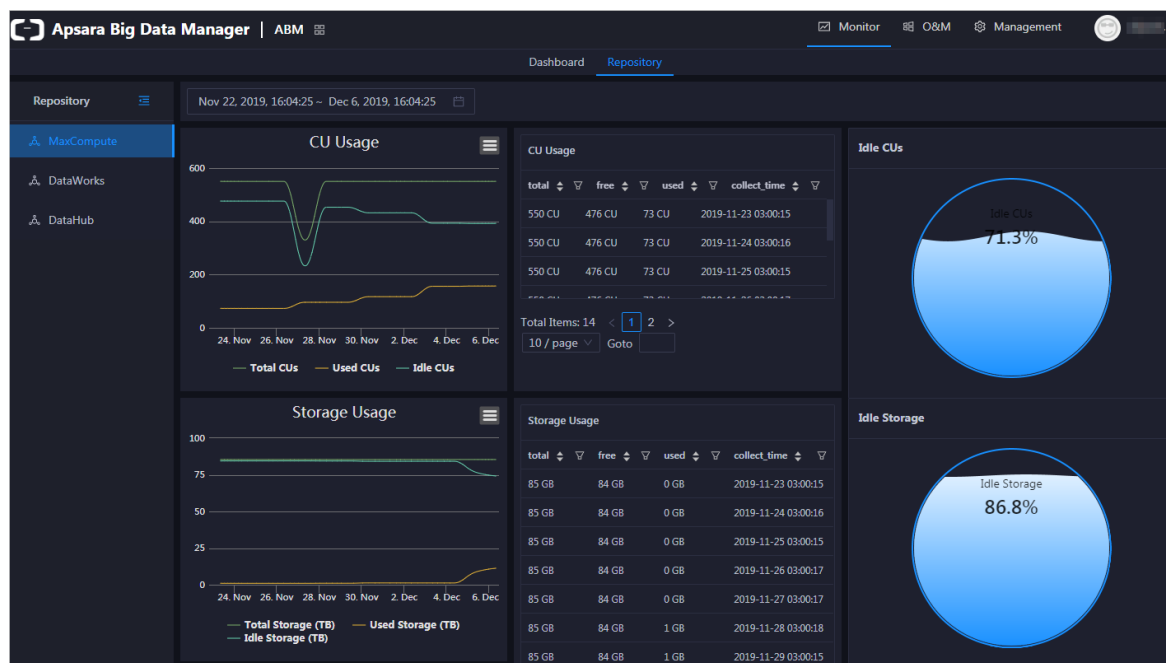
The Repository module displays the resource usage in MaxCompute, DataWorks, and DataHub clusters.

Repository page

1. Log on to the ABM console. The **Dashboard** page appears by default.

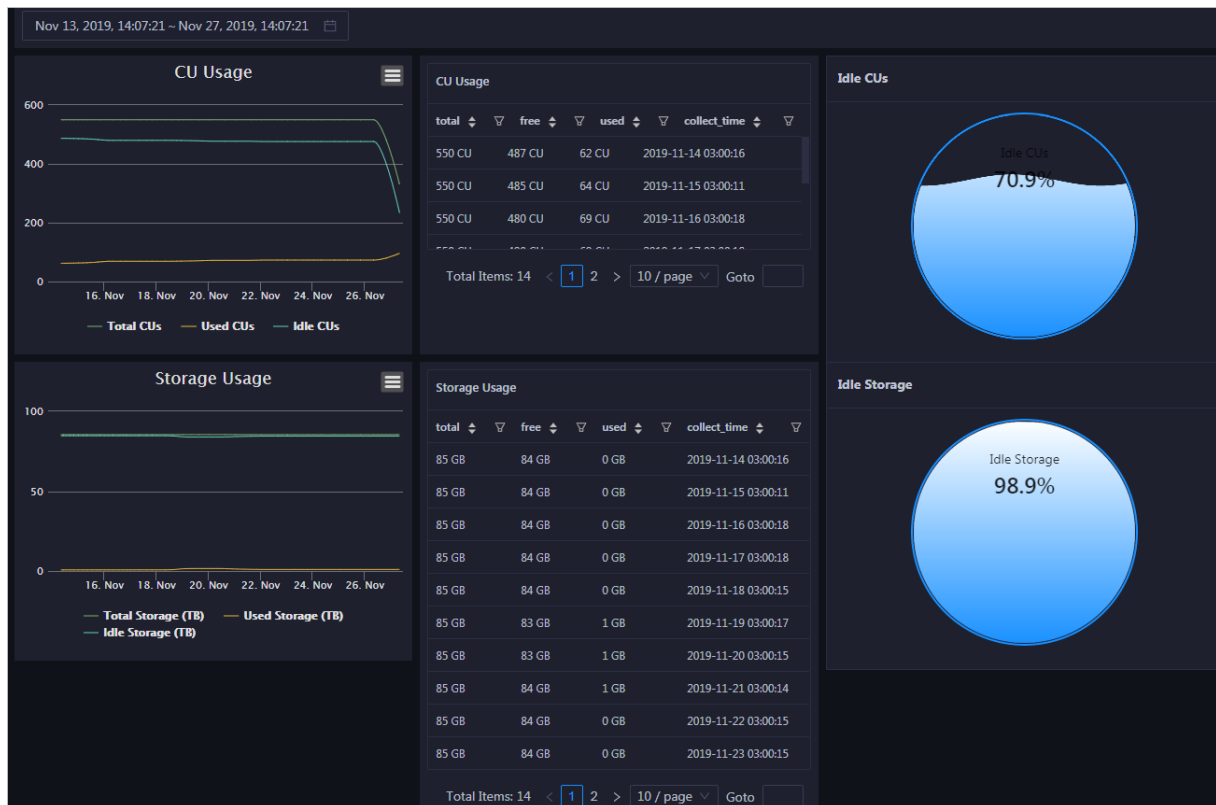
Note After you log on to the ABM console, the **Dashboard** page appears by default. To return to the **Dashboard** page from any other page, click  in the upper-left corner and select **ABM**.

2. On the **Dashboard** page, click the **Repository** tab to go to the **Repository** page.



MaxCompute repository

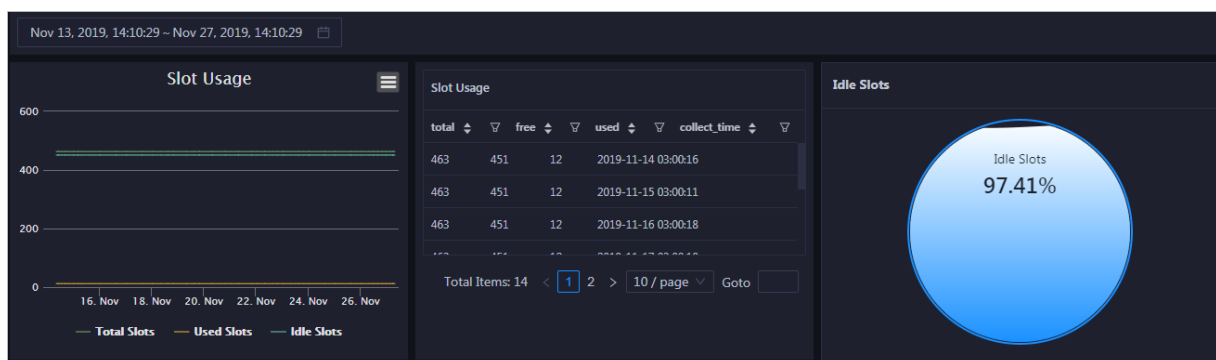
In the left-side navigation pane of the **Repository** page, click **MaxCompute** to view the resource usage in MaxCompute.



This page displays the trends and details of CU and storage usage, and the percentages of idle CUs and storage.

DataWorks repository

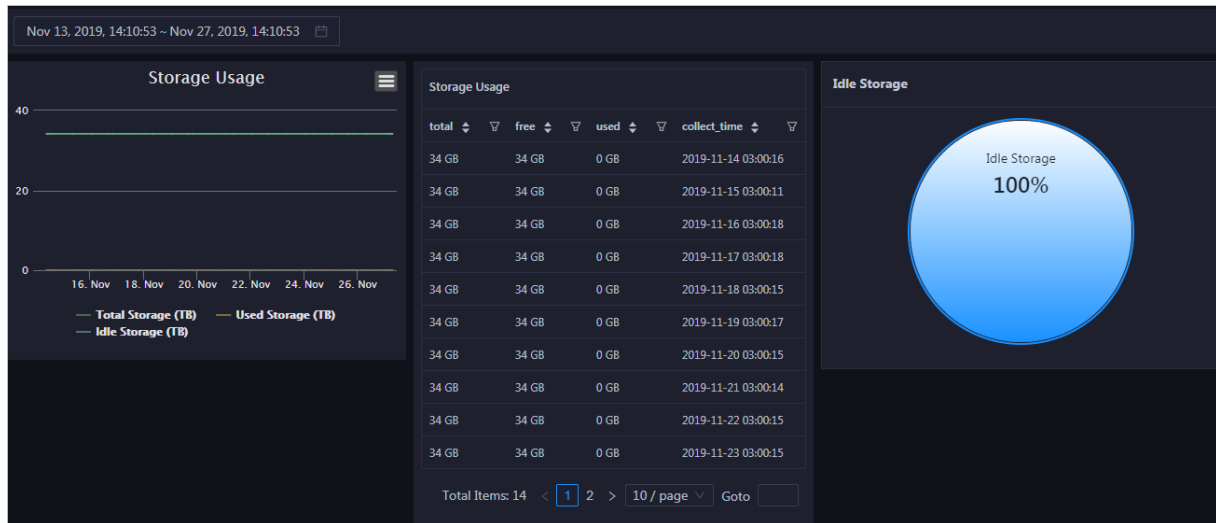
In the left-side navigation pane of the **Repository** page, click **DataWorks** to view the resource usage in DataWorks.



This page displays the trend and details of slot usage, and the percent of idle slots.

DataHub repository

In the left-side navigation pane of the **Repository** page, click **DataHub** to view the resource usage in DataHub.



This page displays the trend and details of storage usage, and the percent of idle storage.

29.4.3. O&M

Apsara Big Data Manager supports O&M management for big data products such as MaxCompute, DataWorks, RealtimeCompute, and DataHub from the business, cluster, service, and host dimensions. For some products, proprietary O&M functions are also deeply customized.

O&M on clusters

O&M on clusters is provided for all big data services. ABM provides two major features for cluster O&M: Overview and Health Status.



- **Overview:** shows the overall running information about a cluster. You can view the host status, service status, health check results, and health check history. You can also view the trend charts of CPU utilization, disk usage, memory usage, load, and packet loss rate for the cluster.
- **Health Status:** shows all check items for a cluster. You can query details of check items and check results for hosts in the cluster. Check results can be CRITICAL, WARNING, or EXCEPTION.

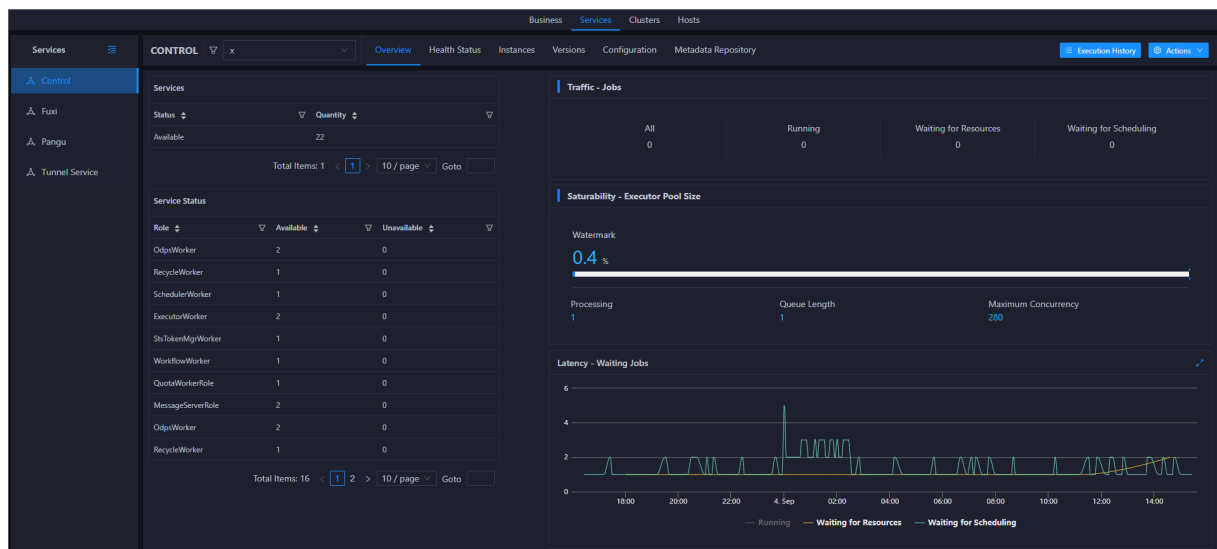
ABM also provides MaxCompute, Realtime Compute for Apache Flink, and DataHub with the following tailored features for cluster O&M:

- Servers: shows information about all the hosts in a cluster. You can view the CPU utilization, memory usage, root disk usage, packet loss rate, and packet error rate.
- Scale in or scale out a cluster: allows you to scale in or scale out a cluster.
- Reverse parse request ID (exclusive for DataHub): allows you to reversely parse a request ID in DataHub to obtain the time that a job is run and the IP address of the host. This helps you query logs for troubleshooting.
- Delete topic from Smoke Testing (exclusive for DataHub): allows you to delete topics from a DataHub test project and view the execution history.

O&M on services

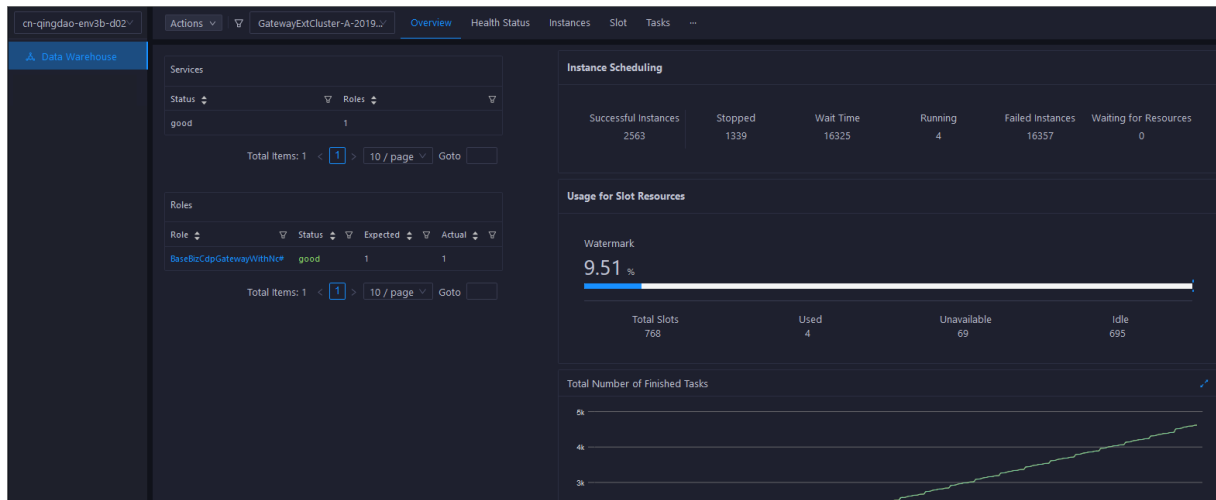
O&M on services is provided for all big data services. ABM provides MaxCompute, DataWorks, DataHub, and Realtime Compute for Apache Flink with tailored features for service O&M.

For MaxCompute, ABM supports O&M on the Control service, Job Scheduler service, Apsara Distributed File System, and Tunnel service.



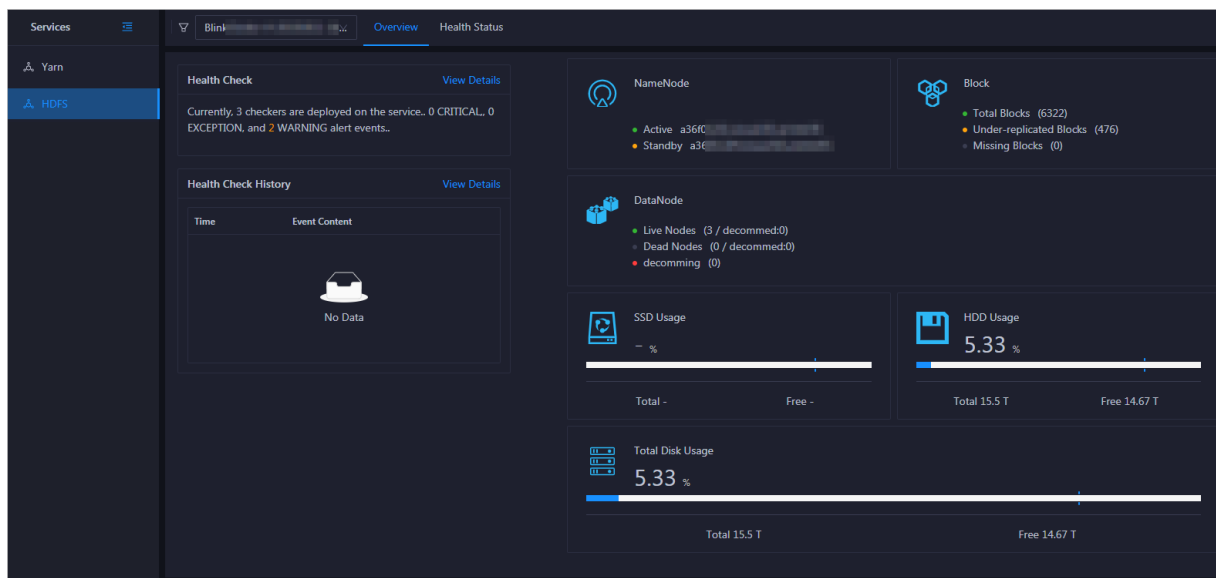
- Control Service: allows you to view the overview, health check items, and instances of the MaxCompute control service. You can configure cluster-level settings and start and stop services.
- Fuxi: shows general information about Job Scheduler, health check items, and instances. You can manage quota groups and add compute nodes to or remove compute nodes from a blacklist or a read-only list. You can also enable or disable SQL acceleration, and restart primary nodes.
- Pangu: shows general information about Apsara Distributed File System, health check items, and instances. You can view the storage information, set the storage node status to disabled or normal, set the disk status to error or normal, and change the primary node. You can also clear the recycle bin, enable or disable data rebalancing, and run checkpoints on primary nodes.
- Tunnel Service: shows general information about the Tunnel service and instances. You can also restart the Tunnel service.

For DataWorks, ABM supports O&M on data warehouses. Data Warehouse shows general information about the services, health status, instances, slots, and service configurations. You can also add or remove hosts to scale out or scale in a cluster.



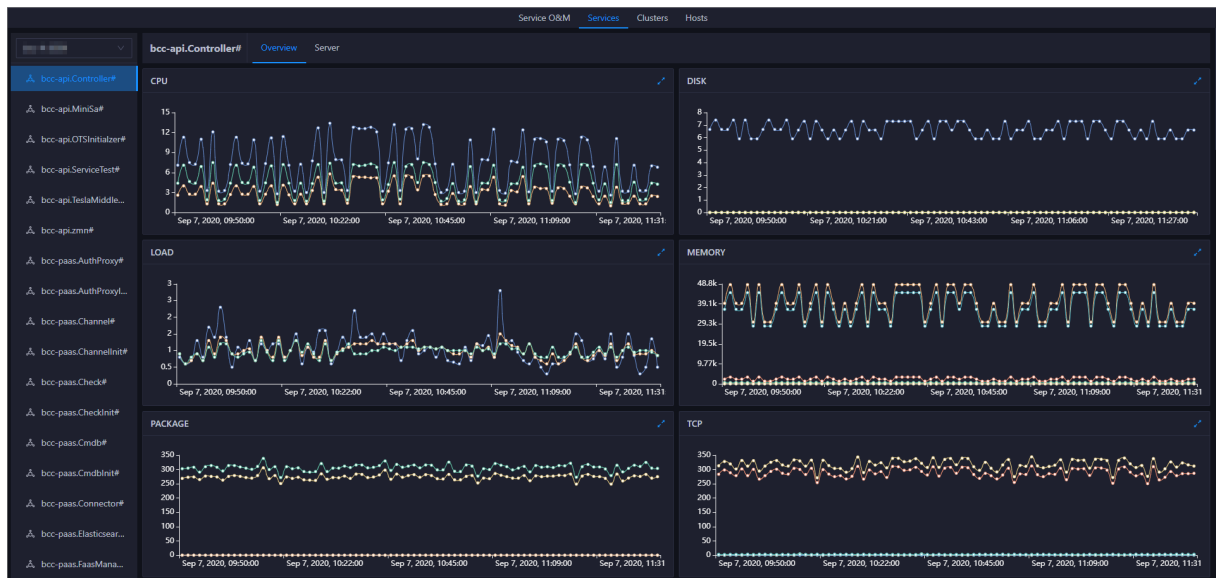
For DataHub, ABM supports O&M on the Control service, Job Scheduler, and Apsara Distributed File System, which are similar to those for MaxCompute.

For Realtime Compute for Apache Flink, ABM supports O&M on Realtime Compute, Yarn, and Hadoop Distributed File System (HDFS).



- Realtime Compute for Apache Flink: shows general information about the Realtime Compute service, service status, health check results, and health check history. You can also view the key metrics of clusters, such as Transaction Processing Systems (TPS) and Failover Rate.
- Yarn: shows information about the YARN service. You can view information about check items, health check results, health check history, applications, containers, and nodes. You can also view logical CPU utilization and logical memory usage.
- HDFS: shows information about the HDFS service. You can view information about check items, health check results, health check history, NameNode, blocks, and DataNode. You can also view Solid State Drives (SSD) usage, Hard Disk Drive (HDD) usage, and total disk usage.

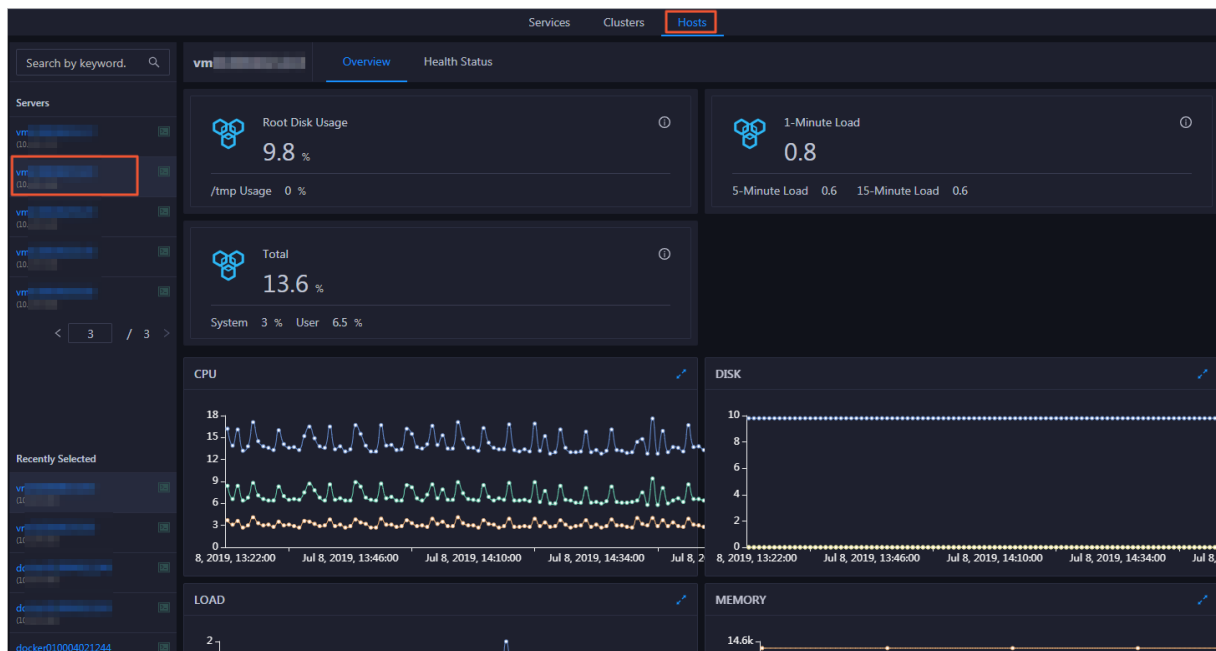
For other big data services, ABM provides information about all server roles in a cluster and the resource usage trend of each server role.



You can select a service from the left-side navigation pane to view the trend charts of CPU utilization, disk usage, memory usage, load, packet loss rate, TCP connections, and root disk usage.

O&M on hosts

O&M on hosts is provided for all big data services. ABM provides two major features for host O&M: Overview and Health Status.



- Overview: shows brief information about hosts in a MaxCompute cluster. You can view the server information, server role status, health check results, and health check history. You can also view the trend charts of CPU utilization, disk usage, memory usage, load, and packet loss rate for the host.
- Health Status: shows all check items for a cluster. You can query details of check items and check results for hosts in the cluster. The check results can be CRITICAL, WARNING, or EXCEPTION.

ABM also provides MaxCompute, DataHub, and Realtime Compute for Apache Flink with the following tailored features for host O&M:

- **Charts:** shows enlarged trend charts of CPU utilization, memory usage, disk usage, load, and packet loss rate for a host. These trend charts are the same as those displayed on the Overview tab for the host.
- **Services:** shows the cluster to which a host belongs, the services running on the host, and the server roles.

O&M on business

ABM provides MaxCompute, Realtime Compute for Apache Flink, Elasticsearch, and DataHub with tailored features for business O&M. For MaxCompute, ABM supports the following features: Projects, Jobs, and Business Optimization.

- **Project management**
 - **Project List:** shows all projects and project details in a MaxCompute cluster. You can filter, query, and sort projects. You can also change the quota group of a project. In the same-city disaster scenario, you can also configuration items the parameters related to destination resource replication and set whether to enable the resource replication function of the project.
 - **Authorize Package for Metadata Repository:** allows you to authorize members of a project to access the metadata warehouse.
 - **Encryption at Rest:** allows you to encrypt the data stored in MaxCompute projects.
 - **Disaster Recovery:** allows you to view the cluster status when zone-disaster recovery is enabled for MaxCompute. You can enable the switchover between the primary and secondary clusters. You can also determine whether to run scheduled tasks to synchronize resources between the primary and secondary clusters.
- **Jobs:** shows information about jobs in a MaxCompute cluster. You can filter and search for jobs. You can also view operational logs, terminate a running job, and collect job logs.
- **Business Optimization:** provides features such as File Merging, File Archiving, and Resource Analysis.

For DataHub, ABM provides information about projects and topics in DataHub clusters.

- **Projects:** shows all projects and project details. Project details are the project overview and related topics.
- **Topics:** shows all topics and topic details. Topic details are the topic overview, information about metrics, shards, subscriptions, DataConnectors, and schemas.

For Realtime Compute for Apache Flink, ABM provides information about projects, jobs, and queues in Realtime Compute for Apache Flink clusters.

- **Projects:** shows all projects.
- **Jobs:** shows all jobs and allows you to diagnose the jobs for troubleshooting.
- **Queues:** shows all queues. The queue resources and jobs in a queue help you analyze queues.

29.4.4. Management

The Management module allows you to manage configurations in a comprehensive manner. This module supports features such as job management, package management, hot update, health management, and operation auditing.

Job management

ABM executes jobs to implement O&M on big data services. Jobs are divided into two types: cron jobs and ordinary jobs. The system executes cron jobs based on a schedule. You can also manually execute cron jobs. Ordinary jobs are all manually executed.

ABM offers multiple O&M schemes to meet the needs of most scenarios. A scheme is a job template. You can easily generate and execute jobs by using a scheme.

ABM also provides an atom library that contains most common O&M operations. An atom is a template of an atomic step. When you generate a job from a scheme, you can directly use atoms as job steps.

Package management

This feature allows you to apply patches to the Docker containers of big data services. Docker is an application container engine. It allows you to quickly update product software by replacing only files that need to be updated.

Hot update

This feature allows you to update monitoring configurations and checkers without the need to interrupt services.

Health management

ABM provides a wide variety of built-in checkers for each big data service. These checkers are used to check service faults and send alerts. This helps you detect and fix faults in a timely manner.

Scheduling: Checkers run scheduling scripts on the servers of specific Tianji roles and generate raw alert data. Raw alert data provides information about the checker, server, alert severity, and alert content. Raw alert data is stored in the ABM database.

Monitoring: You can mount checkers to service pages in ABM. You can configure filter policies to display alerts about high priority checkers.

ABM allows you to customize the execution interval, runtime parameters, and mount point of a checker. You can also enable or disable a checker.

Operation auditing

This feature allows you to view the O&M history and details of all O&M operations. It also allows you to track and identify faults.

29.5. Scenarios

If you are using Apsara Stack and have deployed one or more big data services, you need to use Apsara Big Data Manager (ABM) to perform operations and maintenance (O&M) on these big data services.

Apsara Stack Enterprise and big data services

If you are using Apsara Stack Enterprise and have deployed one or more big data services, such as MaxCompute or DataWorks, you need to use ABM to perform O&M operations on these big data services.

29.6. Limits

None.

29.7. Concepts

This topic describes basic concepts of ABM.

Product

A group of clusters. A product provides services for users.

Cluster

A group of physical hosts. A cluster provides services logically and is used to deploy software of a product. A cluster belongs to only one product. You can deploy multiple services on a cluster.

Service

A group of software used to provide an independent feature. A service contains one or more service roles. You can deploy a service on multiple clusters.

Service role

One or multiple indivisible function units of a service. A service role contains one or more applications. If you deploy a service on a cluster, you must deploy all service roles of the service on hosts in the cluster.

Service role instance

A service role on a specific host. A service role can be deployed on multiple hosts. The service role on a specific host is called a service role instance.

Application

A software entity, which is the minimum unit for starting software. Generally, an application is an executable file or a Docker container. If you deploy a service role on a host, you must deploy all applications of the service role on the host.

Service tree

The overall organizational structure of a product. Each product is an independent entity consisting of a certain number of services. The hierarchy of a product's services forms a service tree.

Workflow

A packaged framework that consists of a sequence of processes predetermined based on specific rules. A workflow supports automatic execution. You can use workflows to perform repetitive tasks.

Job

A product O&M task created by users.

Atom

A template of an atomic step. Atoms can be used to create jobs.

Atomic step

An atom that is directly included as a step when you use schemes to create jobs.

Scheme

A job template. You can use schemes to create jobs.

30.Apsara Uni-manager

30.1. What is Apsara Uni-manager?

Apsara Uni-manager is an enterprise-level management platform designed for both Apsara Stack and hybrid cloud scenarios. Apsara Uni-manager enables delivery, operations, and management of cloud resources. It provides core capabilities such as centralized management, automated O&M, intelligent analysis and customizable scaling. Its simple management style brings excellent user experience that helps enterprises accelerate their digital transformation process.

Components

Apsara Uni-manager consists of the following components:

- Apsara Uni-manager Management Console: provides capabilities such as fine-grained resource management, intelligent data analysis, and customizable scaling by using integrated management portals. This helps enterprises reduce management costs.
- Apsara Uni-manager Operations Console: delivers capacities such as proactive alert and monitoring, root cause locating, and automatic troubleshooting by using automated O&M processes. This can minimize environment maintenance costs and stabilize environments.
- Apsara Uni-manager Dashboards: offers visualized data dashboards for multi-dimensional and panoramic data presentations, including overall running states and resource usage. You can design different dashboards on the homepage for different roles.

Features

Apsara Uni-manager renders centralized management experience, service-oriented operation capabilities, open integration, unified resource monitoring, proactive fault detection, fast fault locating, automatic troubleshooting, intelligent cost analysis, and professional data presentations.

For more information about features, see [Features](#).

Scenarios

Apsara Uni-manager can be used in the following scenarios:

- Multi-level cloud management
- Hybrid cloud management
- Heterogeneous cloud management

For more information about scenarios, see [Scenarios](#).

Benefits

Apsara Uni-manager brings the following benefits:

- Unified portals to maximize user experience
- Intelligent analysis to expedite decision-making
- Open and simple integration

For more information about benefits, see [Benefits](#).

30.2. Benefits

This topic describes the benefits of Apsara Uni-manager and the value for different user roles.

Unified portals to maximize user experience

- Unified portals to enable centralized management and flexible scheduling of hybrid cloud and multi-cloud resources
- Simple self-service experience that is consistent with that on the public cloud
- Integrated O&M from application perspectives

Intelligent analysis to expedite decision-making

- Real-time update of global data and unified resource scheduling
- Automatic fault detection and locating, and automatic troubleshooting suggestions
- Intelligent forecasts for system health early warning

Open and simple integration

- Visualized API portals to effectively reduce learning costs and improve development efficiency
- Standard northbound APIs and SDKs of multiple languages
- Page-level integration and personalized configurations

Value for different user roles

Role	Focus	Value
Decision-maker	Resource presentation	Panoramic resource presentation capabilities, including: • Panoramic data visualization • Regular generation of resource reports • Intelligent recommendations on the homepage • Resource presentation on multiple terminals such as large screens and mobile terminals • Custom analysis presentation capabilities
Operations administrator	Resource operations	Complete hybrid cloud and multi-level cloud operations capabilities, including: • Centralized management of resources and permissions • Automated processing and product upgrades • Forecasts and optimization suggestions for resource quotas • Expansion of operations capabilities such as metering and billing

Role	Focus	Value
Ordinary user	Resource supply	Comprehensive and flexible resource supply capabilities, including: • Centralized activation and management of multi-level cloud resources • Automatic orchestration and elastic scaling of cloud resources • Suggestions on resource allocation and cost optimization • Flexible connection with and use of third-party cloud products • Resource self-service operations for cloud tenants, including activation, operation, and monitoring of resource instances
O&M engineer	Resource O&M	Centralized and efficient O&M capabilities (using automation and intelligent technologies to reduce O&M costs and improve efficiency), including: • Centralized presentation of various types of monitoring data • Automated O&M scripts and tools • Root cause analysis and locating of faults • Fast connection with third-party O&M systems

30.3. Architecture

This topic describes the architecture of Apsara Uni-manager.

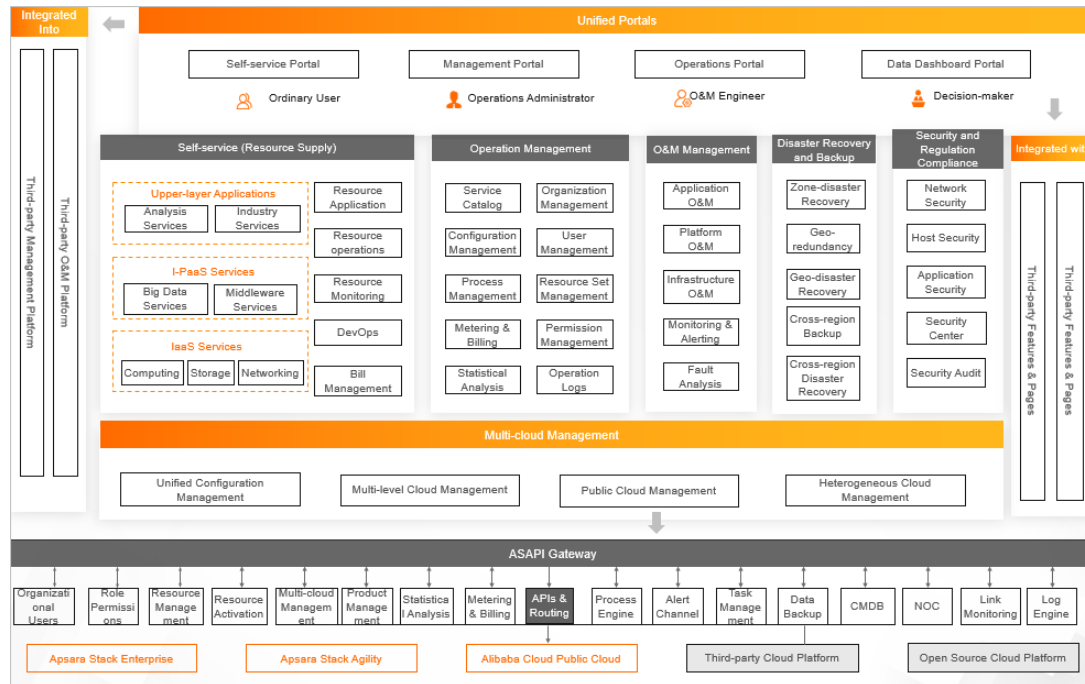
Apsara Uni-manager consists of the following core portals to provide comprehensive cloud management capabilities:

- Self-service portal: provides abilities for ordinary users to autonomously manage all cloud products and cloud resources, such as resource monitoring and resource operations.
- Management portal: provides abilities for operations administrators to manage all cloud products, such as organization management, permission management, metering and billing, statistical analysis, and service management.
- Operations portal: provides O&M engineers with integrated O&M capabilities, such as monitoring & alerting, fault locating, operation changes, and security O&M.
- Data dashboard portal: provides decision-makers with multi-dimensional and panoramic data presentation capabilities. Five dashboards are provisioned: organization resources, network monitoring, security monitoring, alert monitoring, and multi-region resources. Personalized capability expansion is supported.

Apsara Uni-manager also provides multi-cloud management capabilities. Apsara Uni-manager offers standard and open API gateways to third-party integrators. It collects business data and pages from third-party integrators and sends them to multi-cloud environments for integration and adaptation.

The following figure shows the architecture of Apsara Uni-manager.

Apsara Uni-manager architecture



30.4. Features

This topic describes the features of the Apsara Uni-manager Management Console, Apsara Uni-manager Operations Console, and Apsara Uni-manager Dashboards.

Apsara Uni-manager Management Console

The Apsara Uni-manager Management Console provides comprehensive operations capabilities, such as multi-level organization management, role permission management, service management, metering, and billing. The following table describes the features of the Apsara Uni-manager Management Console.

Feature	Description
Centralized management capabilities	Provides centralized resource management capabilities, supports unified user authentication and user permission management, and delivers consistent user experience.
Service-oriented operation capabilities	Provides service management, allows you to enable services by organization, project, and user role, provides standard user-oriented service portals, and supports personalized portal customization.
Open integration	Provides open APIs that use the same standards as the public cloud, supports component expansion for fast connection with existing cloud management systems, and enables personalized expansion of third-party components.

Apsara Uni-manager Operations Console

The Apsara Uni-manager Operations Console provides centralized O&M capabilities and uses automation and intelligent technologies to reduce O&M costs and improve efficiency. The following table describes the features of the Apsara Uni-manager Operations Console.

Feature	Description
Unified resource monitoring	Delivers unified resource monitoring and alert notifications capabilities, and supports centralized configuration of security and alert policies.
Proactive fault detection	Uses machine learning algorithms to render service health check and inventory forecasts, and quickly detects and implements proactive early warning of faults and insufficient inventory based on quick detection and prediction capabilities.
Fast fault locating	Uses the CMDB platform to update inter-resource dependency topology in an automatic and real-time manner, and provides root cause analysis, possible cause screening, and fast fault locating when the preset algorithm engine is used.
Automatic troubleshooting	Uses automated O&M tools to provide an O&M script platform and visualized orchestration capabilities, and delivers automatic troubleshooting methods for a large number of scenarios to reduce manual intervention.
Intelligent cost analysis	Delivers dynamic resource bill analysis to automatically calculate global optimal cost policies, implement cost optimization, and reduce resource waste.

Apsara Uni-manager Dashboards

Apsara Uni-manager Dashboards provides multi-dimensional and panoramic data presentation capabilities. The following table describes the features of Apsara Uni-manager Dashboards.

Feature	Description
Professional data presentation	Provisions five dashboards to display overall running states and resource usage in real time: organization resources, network monitoring, security monitoring, alert monitoring, and multi-region resources.
Personalized customization	Provides an online dashboard editor to support WYSIWYG and customize data sources for personalized customization of business data dashboards.

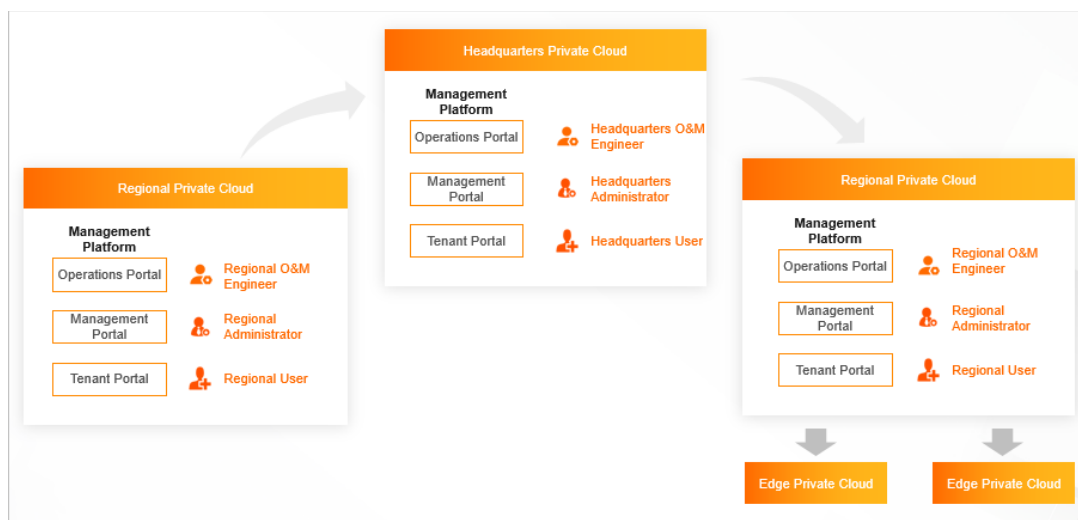
30.5. Scenarios

This topic describes the following scenarios of Apsara Uni-manager: multi-level cloud management, hybrid cloud management, and heterogeneous cloud management.

Multi-level cloud management

Apsara Stack is built at multiple levels and generally includes the headquarters level and regional level. The regional level uses the autonomous management method and is managed by the headquarters level in a centralized manner. O&M engineers can perform O&M at the headquarters level to implement unified multi-level cloud management. Customers can deploy business on different clouds and separately allocate resources to these clouds to improve business experience. The following figure shows the multi-level cloud management scenario.

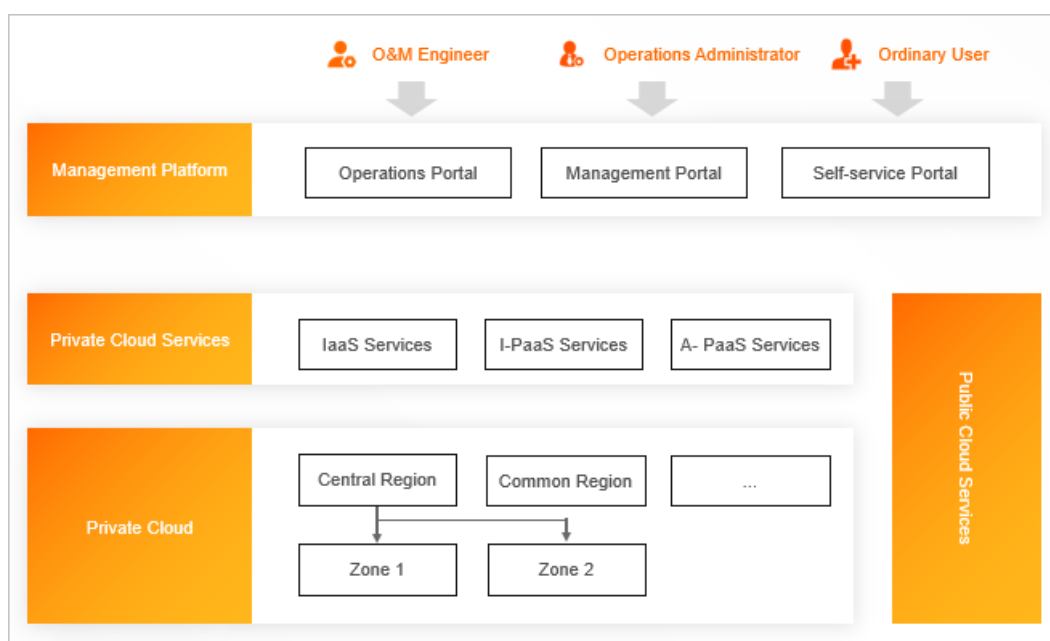
Multi-level cloud management



Hybrid cloud management

Customers have built Apsara Stack and have purchased or plan to purchase a large number of public cloud resources of Alibaba Cloud. They want to centrally manage hybrid cloud resources and implement unified resource supply capabilities. The following figure shows the hybrid cloud management scenario.

Hybrid cloud management



Heterogeneous cloud management

Customers have built private clouds of Alibaba Cloud and other vendors. To expand resource supply capabilities, they must manage multiple heterogeneous clouds in a centralized manner and implement multi-cloud integration. The following figure shows the heterogeneous cloud management scenario.

Heterogeneous cloud management

