ALIBABA CLOUD

# Alibaba Cloud

## Apsara Stack Enterprise

## Product Introduction

Product Version: v3.16.2

Document Version: 20221229

Alibaba Cloud

# Legal disclaimer

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

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:<br>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:<br>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:<br>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:<br>You can use Ctrl + A to select all files. |
| > | Closing angle brackets are used to indicate a multi-level menu cascade. | Click **Settings**> **Network**> **Set network type**. |
| **Bold** | Bold formatting is used for buttons , menus, page names, and other UI elements. | Click **OK**. |
| `Courier font` | Courier font is used for commands | Run the `cd /d C:/window` command to enter the Windows system folder. |
| *Italic* | Italic formatting is used for parameters and variables. | `bae log list --instanceid`<br>*Instance_ID* |
| [] or [a\|b] | This format is used for an optional value, where only one item can be selected. | `ipconfig [-all\|-t]` |
| {} or {a\|b} | This format is used for a required value, where only one item can be selected. | `switch {active\|stand}` |

# Table of Contents

1.Introduction to Alibaba Cloud Apsara Stack 26
1.1. Introduction to Alibaba Cloud Apsara Stack 26
1.1.1. What is Apsara Stack? 26
1.1.2. The reasons to choose Apsara Stack 27
1.1.2.1. Hyper-scale distributed cloud operating system 28
1.1.2.2. Apsara Infrastructure Management 29
1.1.2.3. Centralized cross-region management of multiple da... 29
1.1.2.4. Highly reliable disaster recovery solutions 30
1.1.2.5. Apsara Uni-manager 38
1.1.2.6. OpenAPI 39
1.1.3. Product architecture 39
1.1.3.1. Apsara Stack architecture type 39
1.1.3.2. System architecture 39
1.1.3.3. Network architecture 41
1.1.3.3.1. Overview 41
1.1.3.3.2. Internal network access module 42
1.1.3.3.3. Internet access module 43
1.1.3.3.4. Data exchange module 44
1.1.3.3.5. Integrated access module 45
1.1.3.4. Security architecture 45
1.1.4. Service panorama 46
1.1.5. Scenarios 47
1.1.6. Compliance security solution 49
1.1.6.1. Overview 49
1.1.6.2. Interpretation of key points 49
1.1.6.3. Cloud-based classified protection compliance 50

1.1.6.4. Implementation process of classified protection 53
1.1.6.5. Security compliance architecture 53
1.1.6.6. Benefits 54
2.Apsara Uni-manager 56
2.1. Product Introduction 56
2.1.1. What is Apsara Uni-manager? 56
2.1.2. Benefits 56
2.1.2.1. Apsara Uni-manager Management Console 56
2.1.2.2. Apsara Uni-manager Operations Console 57
2.1.2.3. Apsara Uni-manager Dashboards 58
2.1.2.4. Benefits to different user roles 58
2.1.3. Architecture 59
2.1.4. Features 62
2.1.4.1. Apsara Uni-manager Management Console 62
2.1.4.2. Apsara Uni-manager Operations Console 68
2.1.4.3. Apsara Uni-manager Dashboards 77
2.1.5. Scenarios 79
2.1.5.1. Apsara Uni-manager Management Console 79
2.1.5.2. Apsara Uni-manager Operations Console 80
2.1.5.3. Apsara Uni-manager Dashboards 81
2.1.6. Limits 81
2.1.7. Terms 81
3.Elastic Compute Service (ECS) 82
3.1. Product Introduction 82
3.1.1. What is ECS? 82
3.1.2. Benefits 82
3.1.3. Architecture 84
3.1.4. Features 86

3.1.4.1. Instances 86
3.1.4.1.1. Overview 86
3.1.4.1.2. Instance families 86
3.1.4.1.3. Instance types 88
3.1.4.1.4. UserData 89
3.1.4.1.5. Instance lifecycle 90
3.1.4.1.6. ECS Bare Metal Instance families: 91
3.1.4.1.7. Super Computing Clusters 92
3.1.4.1.8. Dedicated hosts 94
3.1.4.2. Block storage 95
3.1.4.2.1. Overview 95
3.1.4.2.2. Elastic block storage 96
3.1.4.2.2.1. Overview 96
3.1.4.2.2.2. Disks 96
3.1.4.2.2.3. Shared disks 98
3.1.4.2.2.4. Triplicate storage 98
3.1.4.2.2.5. Erasure coding 100
3.1.4.2.3. ECS disk encryption 101
3.1.4.2.4. ECS disk resizing 101
3.1.4.2.5. Local storage 102
3.1.4.3. Images 103
3.1.4.4. Snapshots 104
3.1.4.4.1. Overview 104
3.1.4.4.2. Mechanisms 104
3.1.4.4.3. Specifications of ECS Snapshot 2.0 105
3.1.4.5. Deployment sets 106
3.1.4.6. Network and security 108
3.1.4.6.1. IP addresses of ECS instances of VPC type 108

3.1.4.6.2. ENIs 109
3.1.4.6.3. Internal network 110
3.1.4.6.4. Security group rules 111
3.1.4.7. Tags 111
3.1.5. Scenarios 112
3.1.6. Terms 113
3.2. Introduction to Instance Families and Instance Types 114
3.2.1. Instance families and instance types 114
3.2.2. Intel-based instance families 117
3.2.2.1. Shared instance families 117
3.2.2.2. Dedicated instance families 120
3.2.2.3. ECS Bare Metal Instance families 126
3.2.2.4. Heterogeneous computing and heterogeneous ECS ... 130
3.2.2.5. SCC instance families 132
3.2.2.6. Instance families equipped with local HDDs 133
3.2.2.7. Burstable instance families 134
3.2.3. Hygon-based instance families 135
3.2.3.1. Shared instance families 135
3.2.3.2. Dedicated instance families 135
3.2.3.3. Heterogeneous computing instance families 136
3.2.3.4. Instance families equipped with local HDDs 137
3.2.4. Kunpeng-based instance families 137
3.2.4.1. Shared instance families 137
3.2.4.2. Dedicated instance families 138
3.2.4.3. Instance families equipped with local HDDs 140
3.2.5. Feiteng-based instance families 140
3.2.5.1. Shared instance families 140
3.2.5.2. Dedicated instance families 141

4.Auto Scaling (ESS) 143
4.1. Product Introduction 143
4.1.1. What is Auto Scaling? 143
4.1.2. Benefits 144
4.1.3. Architecture 145
4.1.4. Features 147
4.1.4.1. Scaling groups 147
4.1.4.2. Scaling configurations 148
4.1.4.3. Scaling rules 149
4.1.4.4. Scheduled tasks 149
4.1.4.5. Event-triggered tasks 150
4.1.5. Scenarios 150
4.1.6. Limits 150
4.1.7. Terms 151
5.Container Registry 152
5.1. Product Introduction 152
5.1.1. Container Registry 152
5.1.1.1. Features 152
5.1.1.2. Benefits 153
5.1.1.3. Scenarios 153
6.Container Service for Kubernetes 155
6.1. Product Introduction 155
6.1.1. What is Container Service? 155
6.1.2. Benefits 156
6.1.3. Architecture 157
6.1.4. Features 159
6.1.5. Scenarios 160
6.1.6. Terms 164

7.Resource Orchestration Service (ROS) 167
7.1. Product Introduction 167
7.1.1. What is ROS? 167
7.1.2. Benefits 167
7.1.3. Architecture 168
7.1.4. Features 169
7.1.5. Scenarios 170
7.1.6. Limits 171
7.1.7. Terms 171
8.Object Storage Service (OSS) 172
8.1. Product Introduction 172
8.1.1. What is OSS? 172
8.1.2. Benefits 172
8.1.3. Architecture 173
8.1.4. Terms 174
8.1.5. Features 176
8.1.5.1. Manage buckets 176
8.1.5.1.1. Create a bucket 176
8.1.5.1.2. ACL 177
8.1.5.1.3. Static website hosting 177
8.1.5.1.4. Logging 179
8.1.5.1.5. Lifecycle rules 179
8.1.5.2. Manage objects 181
8.1.5.2.1. Upload objects 181
8.1.5.2.2. ACL 181
8.1.5.2.3. Download objects 182
8.1.5.2.4. Search for objects 182
8.1.5.2.5. Manage objects by using directories 183

8.1.5.2.6. Object tagging 185
8.1.5.3. Data security 187
8.1.5.3.1. Erasure coding 187
8.1.5.3.2. Resource isolation 189
8.1.5.3.3. Disaster recovery 189
8.1.5.3.4. Access permissions and account authorization 190
8.1.5.3.5. Server-side encryption 191
8.1.5.3.6. Client-side encryption 191
8.1.5.4. Data processing 193
8.1.5.4.1. IMG 193
8.1.5.4.2. Video snapshots 194
8.1.6. Scenarios 195
8.1.7. Limits 196
9.Cloud Defined Storage (CDS) 197
9.1. Product Introduction 197
9.1.1. What is CDS? 197
9.1.2. OSS 200
9.1.2.1. Product Introduction 200
9.1.2.1.1. What is OSS? 200
9.1.2.1.2. Benefits 201
9.1.2.1.3. Architecture 202
9.1.2.1.4. Terms 203
9.1.2.1.5. Features 205
9.1.2.1.5.1. Manage buckets 205
9.1.2.1.5.2. Manage objects 210
9.1.2.1.5.3. Data security 216
9.1.2.1.5.4. Data processing 225
9.1.2.1.6. Scenarios 227

9.1.2.1.7. Limits 228
9.1.3. NAS 228
9.1.3.1. Product Introduction 228
9.1.3.1.1. What is NAS? 228
9.1.3.1.2. Benefits 229
9.1.3.1.3. Architecture 230
9.1.3.1.4. Features 230
9.1.3.1.5. Scenarios 231
9.1.3.1.6. Usage notes 232
9.1.3.1.7. Terms 234
9.1.4. EBS 234
9.1.4.1. Product Introduction 234
9.1.4.1.1. Block storage 234
9.1.4.1.2. Terms 236
9.1.4.1.3. EBS 236
9.1.4.1.3.1. Overview 236
9.1.4.1.3.2. Cloud disks 237
9.1.4.1.3.3. Shared disks 238
9.1.4.1.3.4. Features 238
9.1.4.1.4. Local storage 251
9.1.5. Log Service 252
9.1.5.1. Product Introduction 252
9.1.6. CSG 252
9.1.6.1. Product Introduction 252
9.1.6.1.1. What is CSG? 252
9.1.6.1.2. Architecture 253
9.1.6.1.3. Specifications 254
9.1.6.1.4. Features 255

9.1.6.1.5. Scenarios 256
9.1.6.1.6. Usage notes 258
9.1.6.1.7. Reliability and consistency 259
10.Tablestore 261
10.1. Product Introduction 261
10.1.1. What is Tablestore? 261
10.1.2. Benefits 261
10.1.3. Architecture 263
10.1.4. Scenarios 264
10.1.5. Limits 270
10.1.6. Terms 272
10.1.7. Features 273
10.1.7.1. Features 274
10.1.7.2. Tunnel Service 274
10.1.7.3. Global secondary index 275
10.1.7.3.1. Features 275
10.1.7.3.2. Usage notes 276
10.1.7.3.3. Scenarios 278
11.ApsaraDB RDS 287
11.1. Product Introduction 287
11.1.1. Features 287
11.1.2. Benefits 295
11.1.2.1. Ease of use 295
11.1.2.2. High performance 295
11.1.2.3. High security 295
11.1.2.4. High reliability 296
11.1.3. Architecture 297
11.1.4. Features 297

11.1.4.1. Scheduling service 297
11.1.4.2. Data link service 297
11.1.4.3. Instance specification change 298
11.1.4.4. Backup and restoration service 298
11.1.4.5. Monitoring service 299
11.1.4.6. High-availability service 300
11.1.4.7. Migration service 302
11.1.4.8. Dedicated instance family 303
11.1.4.9. Read/write splitting 303
11.1.4.10. Data security 306
11.1.4.11. SQL optimization technology 306
11.1.4.12. SQL audit 310
11.1.5. Scenarios 311
11.1.5.1. Diversified data storage 311
11.1.5.2. Read/write splitting 312
11.1.5.3. Big data analysis 313
11.1.6. Limits 314
11.1.6.1. Limits on ApsaraDB RDS for MySQL 314
11.1.6.2. Limits on ApsaraDB RDS for SQL Server 315
11.1.6.3. Limits on ApsaraDB RDS for PostgreSQL 316
11.1.6.4. Limits on PolarDB 317
11.1.7. Terms 317
11.1.8. Instance types 318
12.KVStore for Redis 341
12.1. Product Introduction 341
12.1.1. What is KVStore for Redis? 341
12.1.2. Enhanced Edition and supported commands 341
12.1.2.1. Performance-enhanced instances of KVStore for Re... 341

12.1.2.2. CAS and CAD commands 345
12.1.2.3. TairString commands 347
12.1.2.4. TairHash commands 357
12.1.2.5. TairGIS commands 380
12.1.2.6. TairBloom commands 388
12.1.2.7. TairDoc commands 394
12.1.3. Benefits 406
12.1.4. System architecture and components 408
12.1.5. Features 409
12.1.6. Scenarios 411
12.1.7. Limits 413
12.1.8. Terms 413
12.1.9. Instance types 414
13.ApsaraDB for MongoDB 421
13.1. Product Introduction 421
13.1.1. ApsaraDB for MongoDB 421
13.1.1.1. What is ApsaraDB for MongoDB? 421
13.1.1.2. Benefits 423
13.1.1.3. Scenarios 425
14.AnalyticDB for PostgreSQL 427
14.1. Product Introduction 427
14.1.1. AnalyticDB for PostgreSQL 427
14.1.1.1. Features 427
14.1.1.2. Benefits 431
14.1.1.3. Scenarios 432
15.Data Transmission Service (DTS) 435
15.1. Product Introduction 435
15.1.1. What is DTS? 435

15.1.2. Benefits 435
15.1.3. Environment requirements 437
15.1.4. Architecture 437
15.1.5. Features 441
15.1.5.1. Data migration 441
15.1.5.2. Data synchronization 452
15.1.5.3. Change tracking 455
15.1.5.4. ETL 457
15.1.5.5. Data consistency 458
15.1.6. Scenarios 459
15.1.7. Concepts 464
16.Data Management (DMS) 467
16.1. Product Introduction 467
16.1.1. What is DMS? 467
16.1.2. Benefits 468
16.1.3. Architecture 469
16.1.4. Features 469
16.1.4.1. Data assets 469
16.1.4.2. SQLConsole 469
16.1.4.3. Database development 471
16.1.4.4. DTS 473
16.1.4.5. Security and specifications 475
16.1.4.6. Solution 476
16.1.4.7. O&M 477
16.1.5. Terms 477
17.Server Load Balancer (SLB) 491
17.1. Product Introduction 491
17.1.1. What is SLB? 491

17.1.2. High availability 492
17.1.3. Architecture 494
17.1.4. Features 496
17.1.5. Scenarios 498
17.1.6. Limits 499
17.1.7. Terms 499
18.Virtual Private Cloud (VPC) 501
18.1. Product Introduction 501
18.1.1. What is a VPC? 501
18.1.2. Benefits 502
18.1.3. Basic architecture 503
18.1.4. Features 504
18.1.5. Use scenarios 505
18.1.6. Background information 507
18.1.7. Limits 507
19.NAT Gateway 510
19.1. Product Introduction 510
19.1.1. What is NAT Gateway? 510
19.1.2. Description 510
19.1.3. Benefits 511
19.1.4. Use scenarios 511
19.1.5. Terms 512
19.1.6. Limits 513
20.Elastic IP Address 515
20.1. Product Introduction 515
20.1.1. EIP overview 515
20.1.2. Limits 516
21.Express Connect 517

21.1. Product Introduction 517
21.1.1. What is Express Connect? 517
21.1.2. Benefits 518
21.1.3. Architecture 518
21.1.4. Scenarios 519
21.1.5. Terms 520
22.VPN Gateway 522
22.1. Product Introduction 522
22.1.1. What is VPN Gateway? 522
22.1.2. Scenarios 523
22.1.3. Limits 525
23.Apsara Stack DNS 527
23.1. Product Introduction 527
23.1.1. What is Apsara Stack DNS? 527
23.1.2. Edition comparison 527
23.1.3. Benefits 530
23.1.4. Architecture 531
23.1.5. Features 533
23.1.6. Scenarios 535
23.1.7. Limits 537
23.1.8. Terms 539
24.Log Service 540
24.1. Product Introduction 540
24.1.1. What is Log Service? 540
24.1.2. Benefits 540
24.1.3. Architecture 541
24.1.4. Features 543
24.1.4.1. Core features 543

24.1.4.2. Other features 544
24.1.4.2.1. Logs 544
24.1.4.2.2. Project 546
24.1.4.2.3. Logstore 546
24.1.4.2.4. Shard 547
24.1.4.2.5. Log topic 549
24.1.5. Scenarios 549
24.1.6. Limits 550
24.1.7. Terms 551
25.API Gateway 553
25.1. Product Introduction 553
25.1.1. What is API Gateway? 553
25.1.2. Architecture 554
25.1.3. Benefits 554
25.1.4. Features 555
25.1.5. Terms 556
26.Message Queue for Apache RocketMQ 559
26.1. Product Introduction 559
26.1.1. What is Message Queue for Apache RocketMQ? 559
26.1.2. Updates 559
26.1.3. Benefits 561
26.1.4. Architecture 562
26.1.5. Functions and features 564
26.1.6. Scenarios 565
26.1.7. Limits 566
26.1.8. Terms 567
27.Apsara Stack Security 571
27.1. Product Introduction 571

27.1.1. What is Apsara Stack Security 571
27.1.2. Benefits 572
27.1.3. Service architecture 574
27.1.4. Features 575
27.1.4.1. On-premises security operations services 575
27.1.4.2. Apsara Stack Security Standard Edition 576
27.1.4.2.1. Overview 576
27.1.4.2.2. Network Detection and Response 577
27.1.4.2.3. Cloud Security Scanner 577
27.1.4.2.4. Server Guard 578
27.1.4.2.5. Security Audit 578
27.1.4.2.6. Web Application Firewall 579
27.1.4.2.7. Threat Detection Service 580
27.1.4.2.8. Security Operations Center 581
27.1.4.3. Optional security services 582
27.1.4.3.1. Overview 582
27.1.4.3.2. Anti-DDoS Service 582
27.1.4.3.3. Sensitive Data Discovery and Protection 583
27.1.4.3.4. Container Protection 583
27.1.5. Restrictions 584
27.1.6. Terms 584
28.MaxCompute 586
28.1. Product Introduction 586
28.1.1. What is MaxCompute? 586
28.1.2. Integration with other Alibaba Cloud services 586
28.1.3. Benefits 589
28.1.4. Architecture 591
28.1.5. Features 594

28.1.5.1. Tunnel 594
28.1.5.1.1. Terms 594
28.1.5.1.2. Tunnel features 595
28.1.5.1.3. Data upload and download through Tunnel 595
28.1.5.2. SQL 596
28.1.5.2.1. Terms 596
28.1.5.2.2. SQL characteristics 596
28.1.5.2.3. Comparison with open source products 596
28.1.5.3. MapReduce 597
28.1.5.3.1. Terms 597
28.1.5.3.2. MapReduce characteristics 597
28.1.5.3.3. MaxCompute MapReduce process 598
28.1.5.3.4. Hadoop MapReduce VS MaxCompute MapReduc... 598
28.1.5.4. Graph 599
28.1.5.4.1. Terms 599
28.1.5.4.2. Graph characteristics 599
28.1.5.4.3. Graph relational network models 599
28.1.5.5. Unstructured data processing in integrated comput... 600
28.1.5.6. Unstructured data processing in MaxCompute 601
28.1.5.7. Enhanced features 601
28.1.5.7.1. Spark on MaxCompute 601
28.1.5.7.1.1. Terms 601
28.1.5.7.1.2. Features of Spark on MaxCompute 601
28.1.5.7.1.3. Spark features 602
28.1.5.7.1.4. Spark architecture 602
28.1.5.7.1.5. Benefits of Spark on MaxCompute 603
28.1.5.7.2. Elasticsearch on MaxCompute 604
28.1.5.7.2.1. Overview 604

28.1.5.7.2.2. Features of Elasticsearch on MaxCompute 604
28.1.5.7.2.3. Elasticsearch features 605
28.1.5.7.2.4. Elasticsearch architecture 606
28.1.5.7.2.5. Benefits 607
28.1.5.8. Mlulti-region deployment 608
28.1.5.8.1. MaxCompute multi-region deployment 608
28.1.6. Scenarios 609
28.1.6.1. Scenario 1: Migrate data to the cloud cost-effective... 609
28.1.6.2. Scenario 2: Improve development efficiency and re... 609
28.1.6.3. Scenario 3: Use mass data to achieve precision m... 610
28.1.6.4. Scenario 4: Achieve precision marketing with big d.. 610
28.1.7. Limits 611
28.1.8. Terms 611
28.1.9. Storage performance 614
29.Realtime Compute 618
29.1. Product Introduction 618
29.1.1. What is Realtime Compute? 618
29.1.2. End-to-end real-time computing 619
29.1.3. Differences between real-time computing and batch c... 620
29.1.3.1. Overview 620
29.1.3.2. Batch computing 620
29.1.3.3. Real-time computing 621
29.1.3.4. Comparison between real-time computing and bat... 622
29.1.4. Benefits 623
29.1.5. Product architecture 624
29.1.5.1. Workflow 624
29.1.5.2. Business architecture 627
29.1.5.3. Technical architecture 627

29.1.6. Features 629
29.1.7. Product positioning 631
29.1.8. Scenarios 632
29.1.8.1. Overview 632
29.1.8.2. Management of e-commerce activities 632
29.1.8.3. Multidimensional analysis of data from IoT sensors 633
29.1.8.4. Big screen service for the Tmall Double 11 Shoppi... 637
29.1.8.5. Mobile data analysis 638
29.1.9. Restrictions 639
29.1.10. Basic concepts 639
30.DataHub 641
30.1. Product Introduction 641
30.1.1. What is DataHub? 641
30.1.2. Benefits 641
30.1.3. Architecture 642
30.1.4. Features 643
30.1.4.1. Data queue 643
30.1.4.2. Checkpoint-based data restoration 643
30.1.4.3. Data synchronization 643
30.1.4.4. Scalability 644
30.1.5. Scenarios 644
30.1.5.1. Overview 644
30.1.5.2. Data uploading 645
30.1.5.3. Data collection 645
30.1.5.4. Realtime Compute 645
30.1.5.5. Data utilization 646
30.1.5.6. Data archiving 646
30.1.6. Limits 646

30.1.7. Terms 647
31.DataWorks 650
31.1. Product Introduction 650
31.1.1. What is DataWorks? 650
31.1.2. Benefits 650
31.1.3. Architecture 651
31.1.3.1. Service architecture 652
31.1.3.2. System architecture 652
31.1.3.3. Security architecture 652
31.1.3.4. Multitenancy 653
31.1.4. Services 653
31.1.4.1. Data Integration 653
31.1.4.2. DataStudio 655
31.1.4.2.1. Overview 655
31.1.4.2.2. Workflows 655
31.1.4.2.3. Solutions 655
31.1.4.2.4. Code editor 656
31.1.4.2.5. Code repository and team collaboration 657
31.1.4.3. Administration 657
31.1.4.3.1. Overview 657
31.1.4.3.2. Overview page 657
31.1.4.3.3. Node O&M pages 657
31.1.4.3.4. Intelligent Monitor service 658
31.1.4.3.5. Engine O&M 658
31.1.4.4. DataAnalysis 658
31.1.4.5. Data Map 659
31.1.4.6. Security Center 659
31.1.4.7. DataService Studio 660

31.1.4.8. Migration Assistant 660
31.1.4.9. Workspace Management 661
31.1.4.10. Data Asset Management 662
31.1.4.11. Data Protection 663
31.1.4.11.1. Overview 663
31.1.4.11.2. Terms 663
31.1.4.11.3. Management 664
31.1.4.11.4. Data recognition 664
31.1.4.11.5. Data Activities 664
31.1.4.11.6. Data masking 665
31.1.4.11.7. Levels 665
31.1.4.11.8. Manual check 665
31.1.4.11.9. Data risks 665
31.1.4.11.10. Risk Rules 665
31.1.4.11.11. Data Auditing 665
31.1.5. Scenarios 666
31.1.5.1. Cloud-based data warehouse 666
31.1.5.2. Business intelligence 666
31.1.5.3. Data-driven management 666
31.1.6. Limits 667
31.1.7. Terms 667
32.Apsara Big Data Manager (ABM) 671
32.1. Product Introduction 671
32.1.1. What is Apsara Big Data Manager? 671
32.1.2. Benefits 671
32.1.3. Architecture 671
32.1.3.1. O&M Architecture 672
32.1.4. Features 673

32.1.4.1. Dashboard 673
32.1.4.2. Repository 676
32.1.4.3. O&M 678
32.1.4.4. Management 683
32.1.5. Scenarios 683
32.1.6. Limits 683
32.1.7. Concepts 684

# 1.Introduction to Alibaba Cloud Apsara Stack

## 1.1. Introduction to Alibaba Cloud Apsara Stack

### 1.1.1. What is Apsara Stack?

Apsara Stack is an open, unified, and trusted full-stack cloud platform tailored for enterprise customers. Apsara Stack is developed based on the same distributed architecture as Alibaba Cloud public cloud. Apsara Stack allows enterprise customers to deploy Alibaba Cloud services in on-premises environments, easily integrate with the public cloud, and enjoy hybrid cloud services anytime and anywhere.

#### Private clouds

A private cloud is a cloud computing system that is deployed on premises for public service sectors and enterprises by cloud computing service providers. The cloud infrastructure, software, and hardware resources of a private cloud are deployed behind a firewall to allow internal departments of an enterprise to share the resources of a data center. 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 the enterprise scale or business requirements:

- Multi-tenant comprehensive private clouds for industries and large groups: full-stack cloud systems 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 for production environments.
- Single-tenant basic private clouds for small and medium-sized enterprises and scenarios: cloud systems 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

As enterprises transform their IT infrastructure to be cloud-based, they must consider construction requirements such as security compliance, reuse of existing data centers, and superb experience of on-premises systems. Increased enterprises prefer to use their own data centers to deliver a service experience that relies on large-scale cloud computing.

Apsara Stack is an extension of Alibaba Cloud public cloud and brings public cloud technologies to private clouds. Apsara Stack delivers a complete and customizable software solution based on Alibaba Cloud and allows enterprises to deploy the hyper-scale cloud computing and big data services of Alibaba Cloud public cloud within their own data centers. Apsara Stack provides enterprises with a consistent hybrid cloud experience. Apsara Stack helps ensure business continuity by allowing users to obtain IT resources on demand.

Apsara Stack supports on-premises deployment and can be independently run and managed outside Alibaba Cloud.

## Benefits

Apsara Stack helps public service sectors and enterprises digitally transform their business systems and services based on the abundant services and digitalization practices of Alibaba Group in combination with mature solutions developed from experience. Apsara Stack provides the following benefits:

- Elastic

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

- Agile

  Uses Internet and microservices integration to accelerate innovation.

- Ultra large

  Allows enterprises to deploy over 10,000 servers within a single region with multi-tenant support. Apsara Stack allows enterprises to deploy services across multiple regions and is capable of handling tremendous volumes of business requests.

- Secure and reliable

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

- Online operable

  Provides industry-level cloud services that support accurate metering and billing. Apsara Stack allows users to use services on demand and easily manage services.

## Characteristics

Apsara Stack is an enterprise-class cloud platform with the following characteristics:

- Software-defined platform: hides underlying hardware differences and supports horizontal or vertical scaling based on resource requirements without the awareness of upper-layer applications.
- Production-level reliability and security compliance: ensures the continuity and security of enterprise data.
- Centralized access management: controls user permissions by using different roles to facilitate O&M.

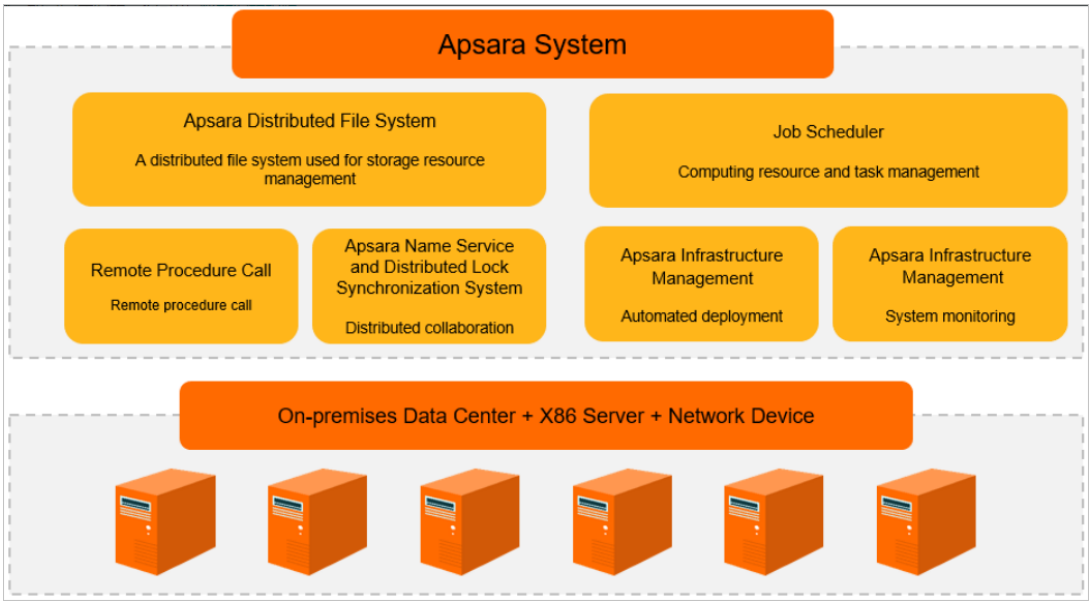
# 1.1.2. The reasons to choose Apsara Stack

# 1.1.2.1. Hyper-scale distributed cloud operating system

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

Apsara Distributed Operating System is a hyper-scale universal operating system developed by Alibaba Cloud. Apsara Distributed Operating System connects millions of servers around the world to act as a supercomputer and provides computing capabilities as online public services. The computing capabilities provided by Apsara Distributed Operating System are powerful, universal, and accessible to everyone.

Apsara Distributed Operating System architecture




Apsara Distributed Operating System consists of the following modules:

- Underlying services for distributed systems

  This module provides the coordination, remote procedure call (RPC), 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 load balancing. An API similar to Portable Operating System Interface of UNIX (POSIX) is provided to access files in the user space. 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 status and performance metrics of upper-layer application services, and generates alerts and records of exception events. Additionally, the module provides O&M engineers with deployment and configuration management of the entire Apsara system and its upper-layer applications. The module supports both the online elastic scaling of clusters and the online upgrade of application services.

## 1.1.2.2. Apsara Infrastructure Management

Apsara Infrastructure Management provides cloud services with basic support capabilities such as centralized deployment, authentication, authorization, and control.

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

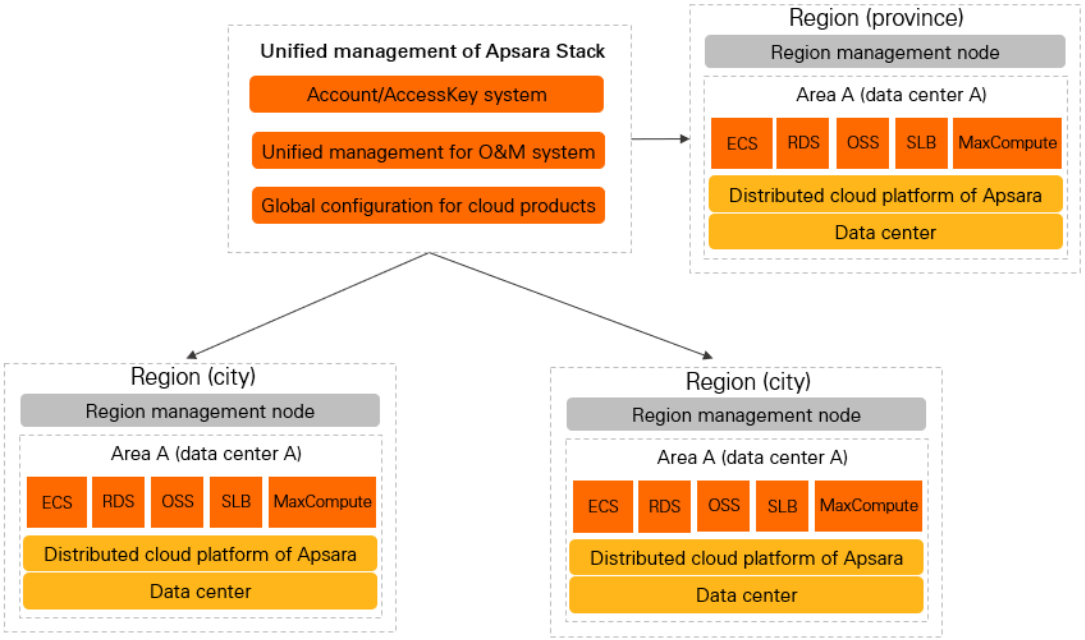
- The deployment framework provides an access platform for centralized service deployment and manages service dependencies.
- The resource library stores the executable files of all cloud services and components on which the cloud services depend.
- The authentication and authorization module provides 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.

## 1.1.2.3. 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




# 1.1.2.4. Highly reliable disaster recovery solutions

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

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 business continuity.

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.

ASR-BR 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, ASR-BR can use the backup data to recover businesses. This ensures business continuity and data security.

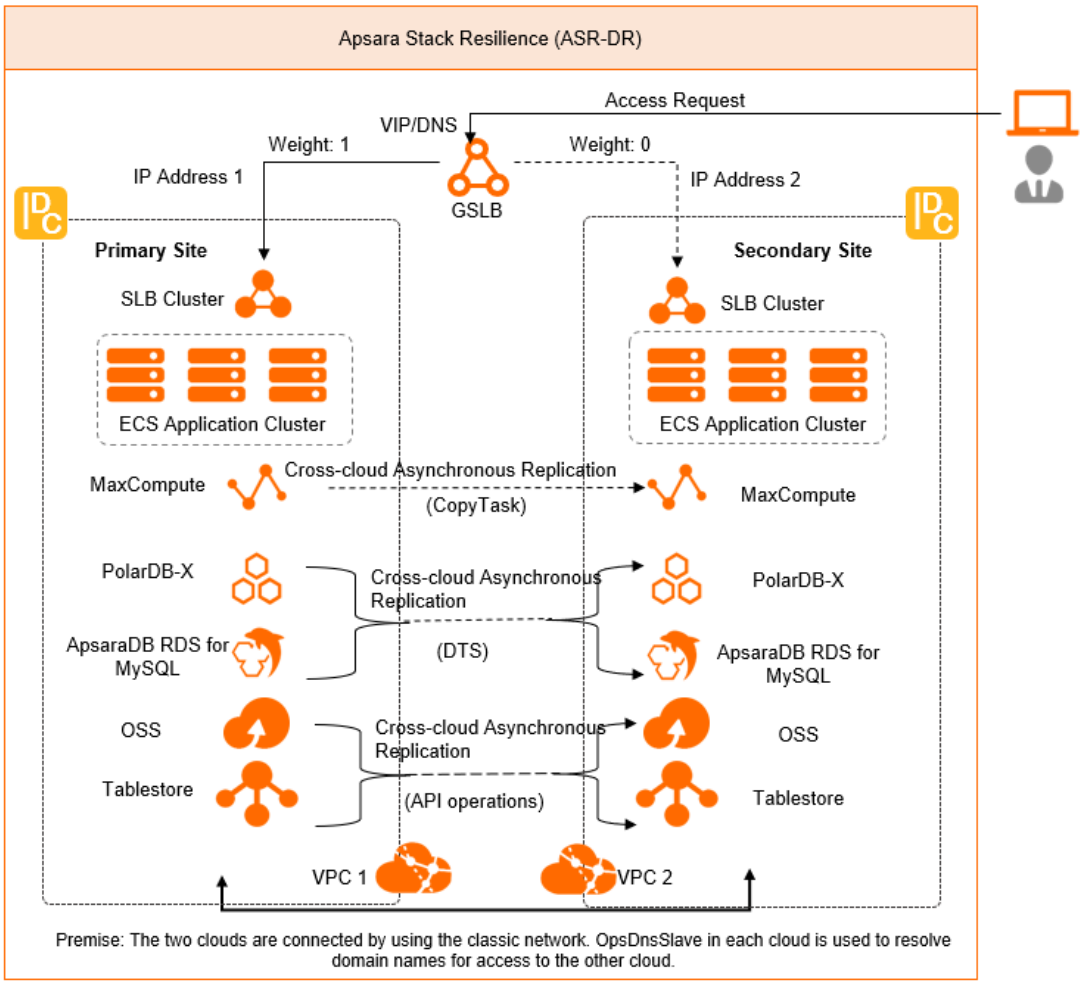
## Geo-disaster recovery

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.

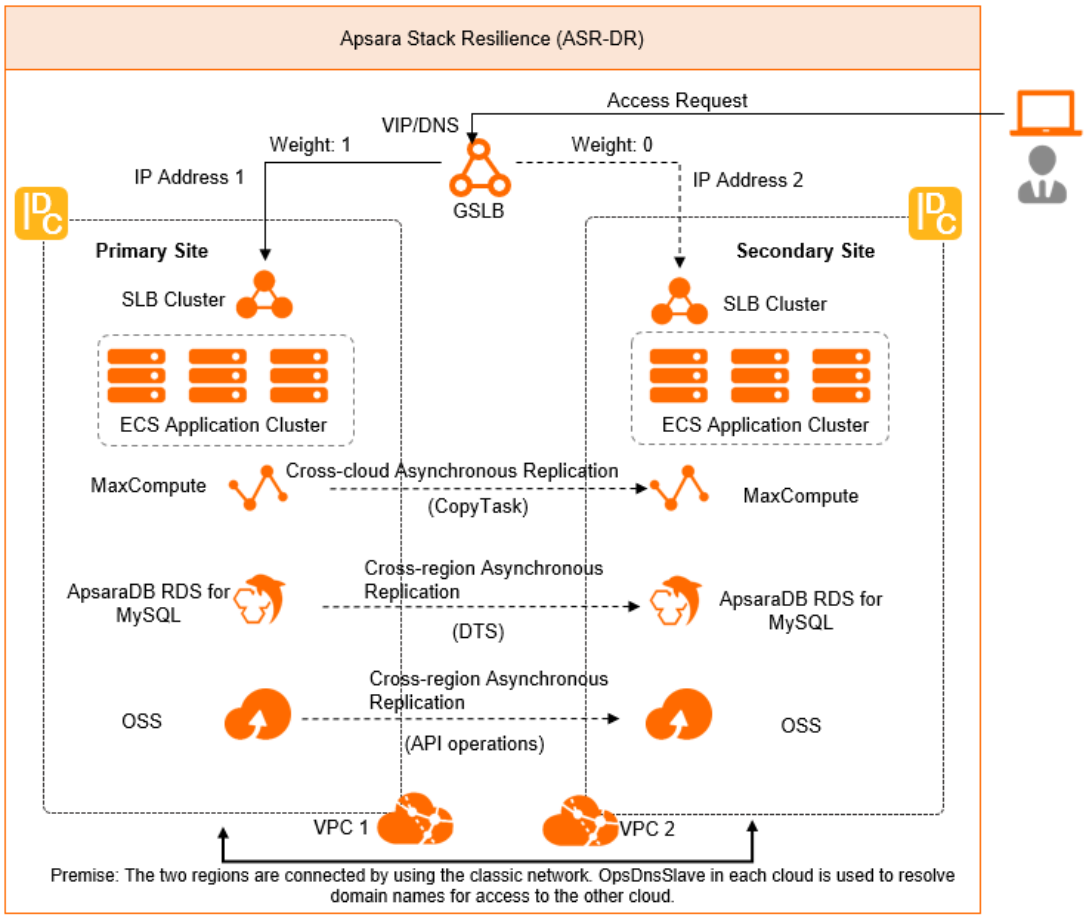
- Cross-cloud DR is implemented between two cloud instances. The primary and secondary sites are two independent cloud instances deployed in different locations and use independent account systems. Users must use separately authorized accounts to log on to the two cloud instances.
- Cross-region DR is 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

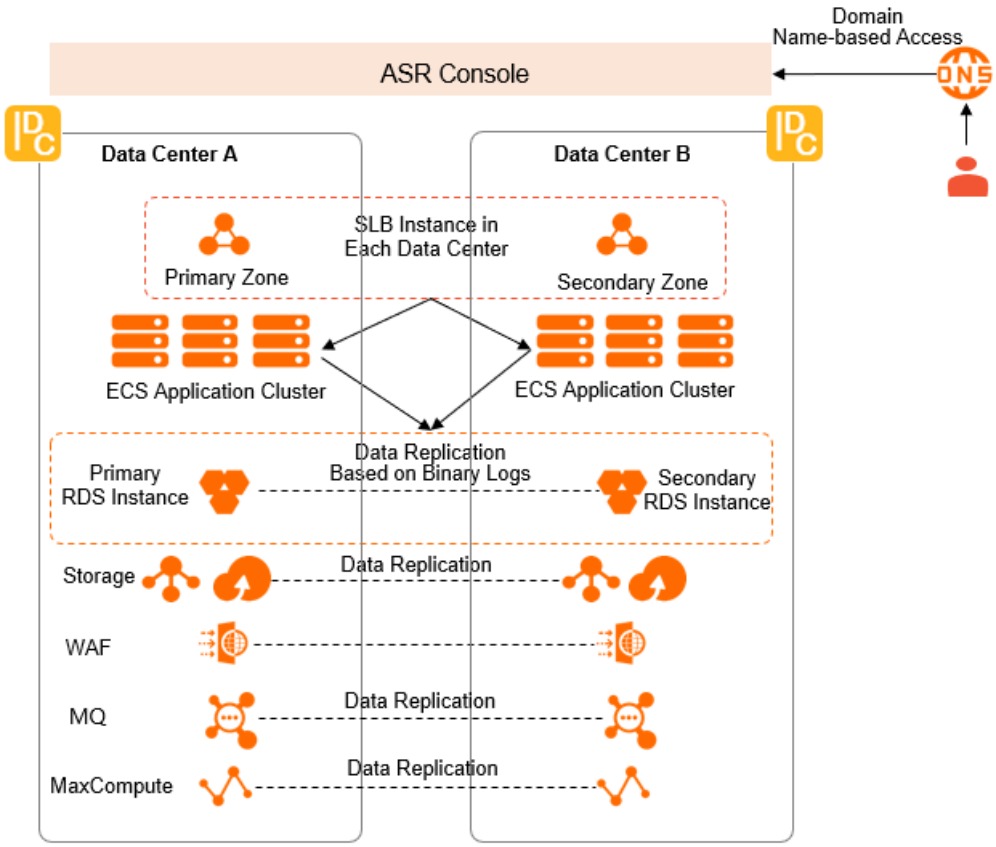
Architecture of cross-region DR




## Zone-disaster recovery

Zone-disaster recovery refers to two independent, mutually backed-up data centers within the same region. If 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

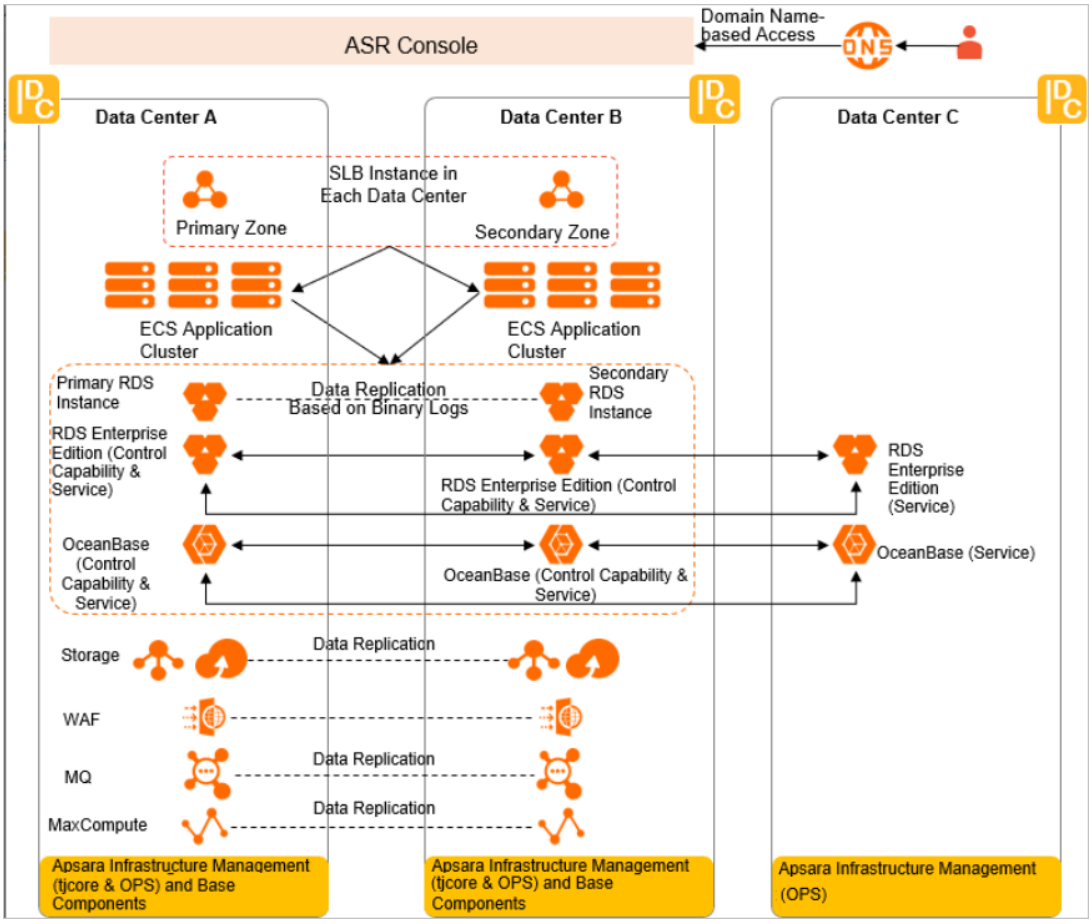



**Note** Data Center A is the primary data center. Data Center B is the secondary data center.

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 if 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



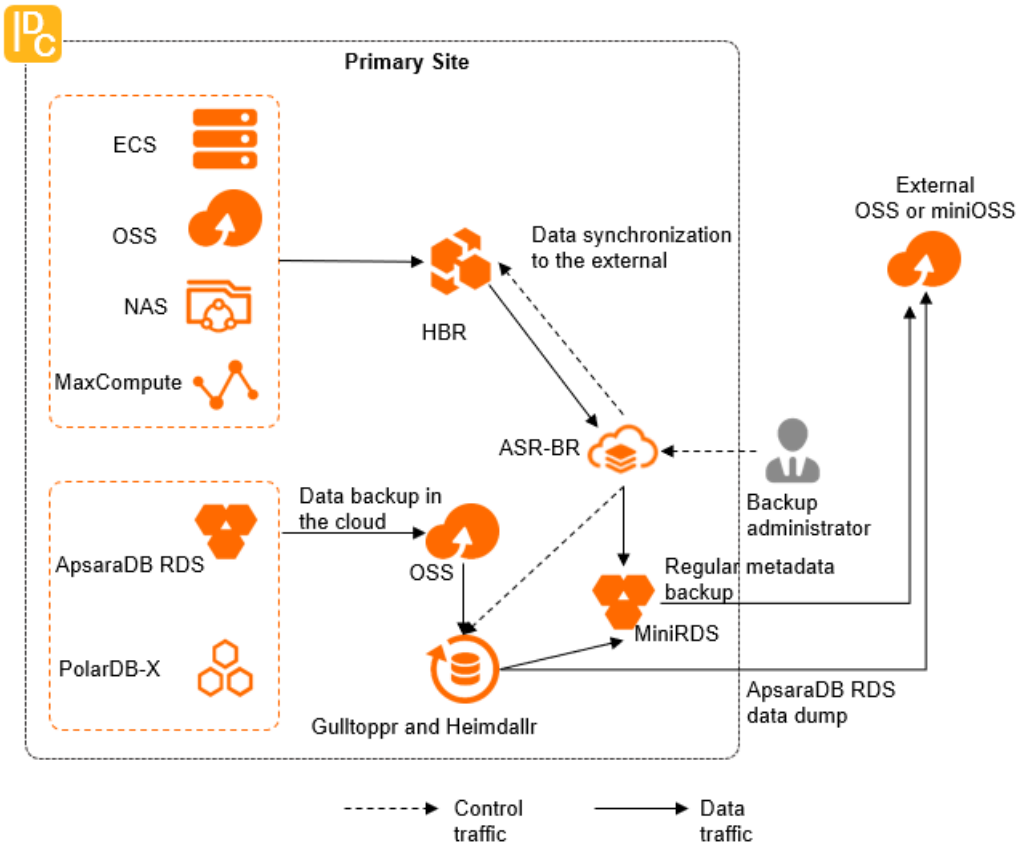

## ASR-BR

ASR-BR supports the following two architecture modes: local backup and cross-region backup.

- Local backup

  Data of the primary site is backed up to an OSS-compliant storage system on a regular basis. You must deploy ASR-BR and related services at the primary site. This architecture reduces deployment costs.

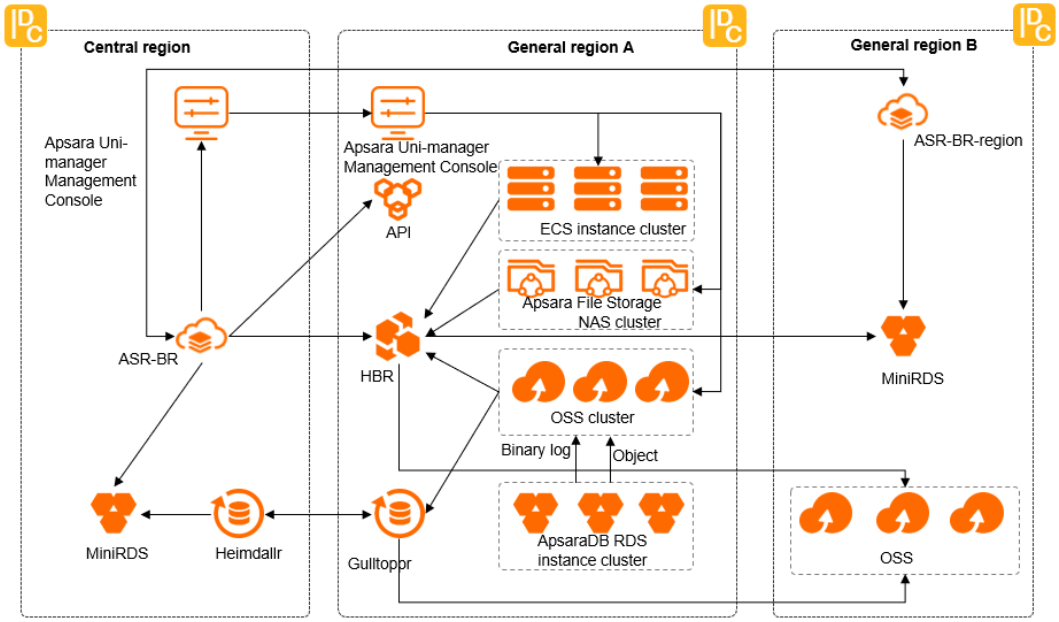
  Architecture of local backup




- Cross-region backup

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

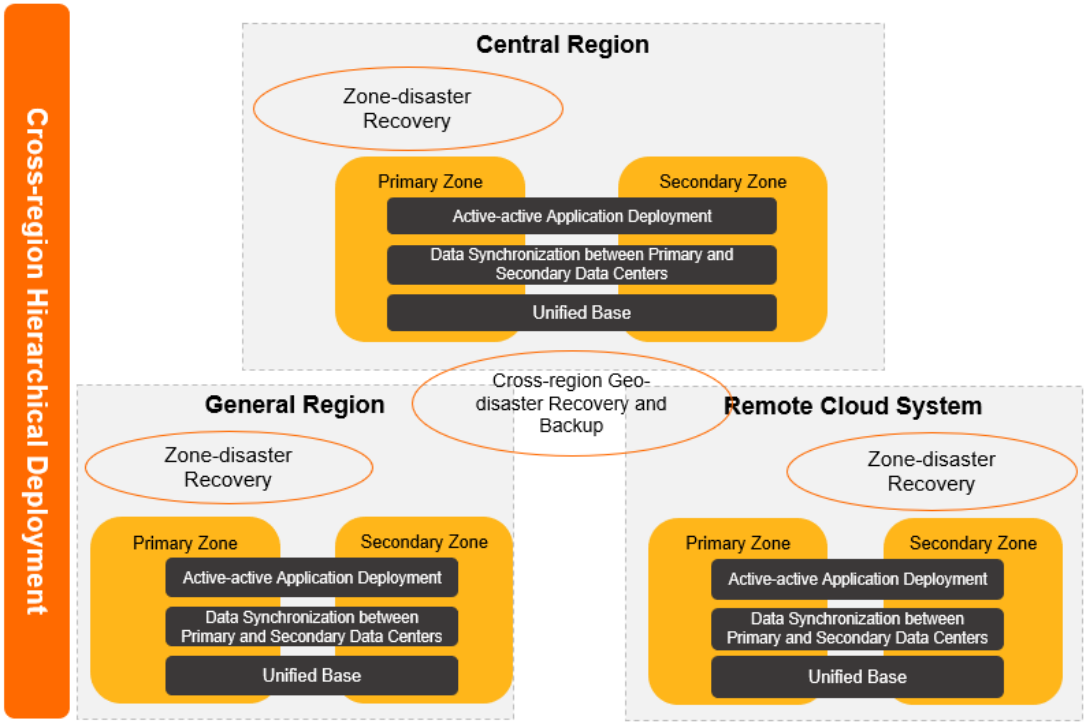
  Architecture of cross-region backup




## 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



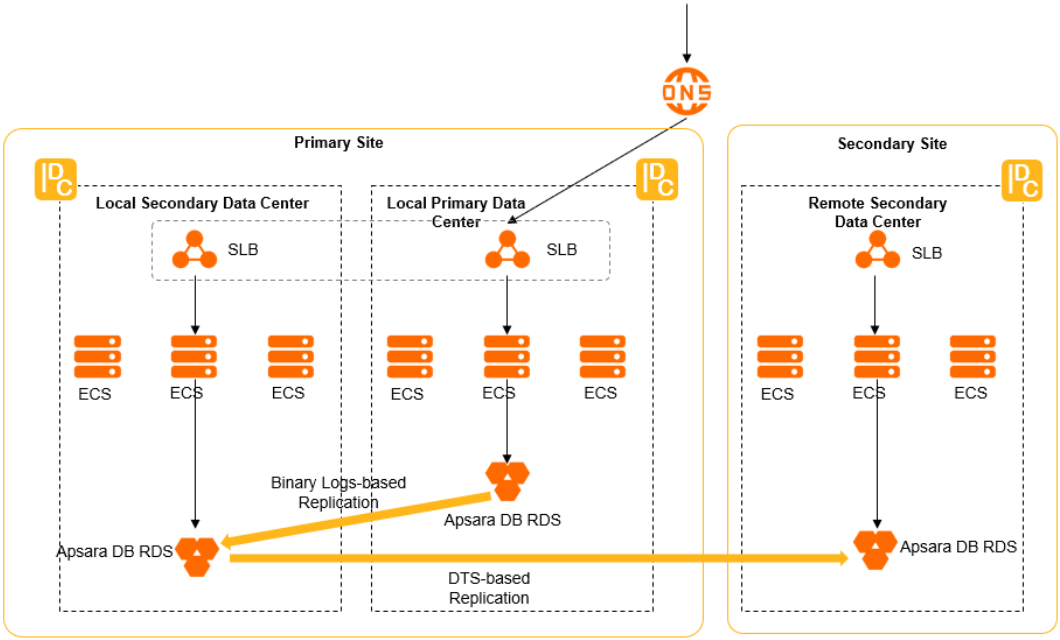

## Three data centers across two zones

In the DR solution of three data centers across two zones, two zones include the local and remote zones. Three data centers include the local primary data center, the 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 ApsaraDB RDS and OSS.

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

  ApsaraDB RDS is independently deployed at each of the primary and secondary sites. Two data centers are deployed at the primary site. ApsaraDB RDS is independently deployed in each of the two data centers. The primary and secondary ApsaraDB 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 ApsaraDB RDS instance.

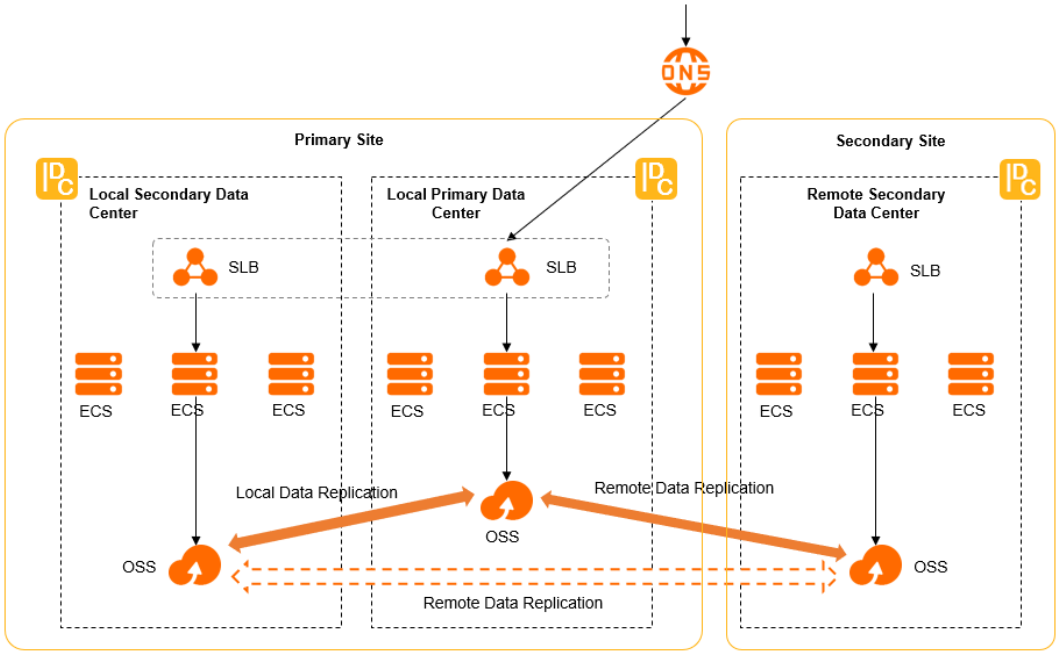
  Architecture of ApsaraDB RDS deployed in three data centers across two zones




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

  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 zones



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 few clicks to ensure business continuity. If both data centers at the primary site fail, the BDR administrator initiates a failover plan in the 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 bucket at the primary site and the secondary bucket at the secondary site. 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 failback plan to fail back the traffic to the primary site.

# 1.1.2.5. Apsara Uni-manager

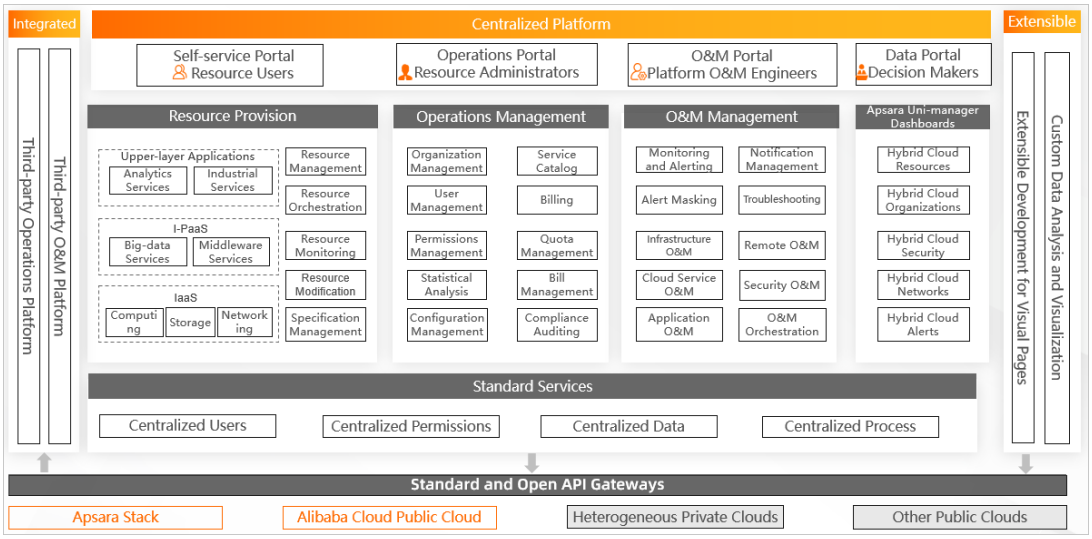
This topic describes the components and benefits of Apsara Uni-manager.

Apsara Uni-manager is an enterprise-class cloud management platform provided by Apsara Stack. Apsara Uni-manager can be used in Apsara Stack and hybrid cloud scenarios. Apsara Uni-manager supports provisioning, operations, and management of cloud resources. Apsara Uni-manager provides core capabilities such as centralized management, intelligent O&M, fine-grained operations, and custom extensions. Apsara Uni-manager simplifies hybrid cloud management, improves user experience, and helps enterprises accelerate digital transformation.

Apsara Uni-manager consists of the following components:

- Apsara Uni-manager Management Console: provides an integrated management portal that delivers capacities such as fine-grained resource governance, intelligent data analysis, and custom feature extensions. This can minimize cloud management costs.
- Apsara Uni-manager Operations Console: uses automated O&M processes to deliver capacities such as proactive alerting and monitoring, root cause locating, and automatic troubleshooting. This can minimize environment maintenance costs and stabilize environments.
- Apsara Uni-manager Dashboards: provides visualized data dashboards for multi-dimensional and panoramic data presentations, including overall status and resource usage of hybrid clouds. You can design different dashboards on the homepage for different roles.

Schematic diagram of Apsara Uni-manager



Apsara Uni-manager brings the following benefits:

- Unified entry

  Apsara Uni-manager provides a unified entry that consists of a self-managed portal, an operations portal, an O&M portal, and a data portal. This delivers a comprehensive range of cloud management capabilities for users in different businesses.

- Centralized management

  Apsara Uni-manager provides centralized management of users, permissions, data, and processes.

- Openness for easy integration and expansion

  Apsara Uni-manager can manage multiple clouds by using open and centralized API Gateway. Third-party data and webpages are collected by using northbound APIs and delivered to multi-cloud environments for integration by using southbound APIs.

## 1.1.2.6. 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.

# 1.1.3. Product architecture

## 1.1.3.1. Apsara Stack architecture type

This topic describes the architecture type of Apsara Stack.

Apsara Stack adopts a native cloud architecture and is built on Alibaba Cloud-developed operating system, distributed technologies, and products. This single architecture supports all cloud services and allows the complete openness of the cloud platform. Apsara Stack comes with comprehensive service features for enterprises, delivers disaster recovery and backup capabilities, and can be fully self-managed.

## 1.1.3.2. System architecture

This topic describes the system architecture and logical architecture of Apsara Stack, and the features of each layer.

As shown in System architecture of Apsara Stack, the system architecture of Apsara Stack consists of the following layers:

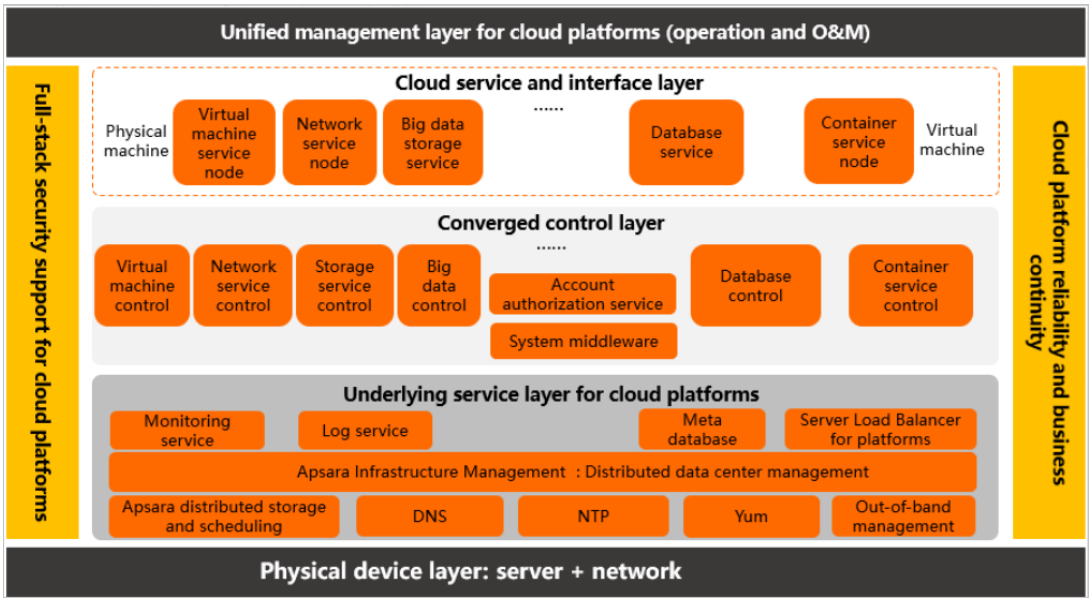
- Physical device layer: includes hardware devices for cloud computing, such as servers and network devices.
- Basic service layer: provides the basic service capabilities, including out-of-band management, system cloning, clock source, YUM source, metadatabase, and platform log service.
- Converged management and control layer: manages and controls a variety of cloud services on Apsara Stack.
- Cloud service and API layer: provides centralized management and O&M for VMs and physical

machines by using a converged node management mechanism, and provides an open API platform for centralized API management and custom development.

- Centralized management layer: provides a unified entry for centralized O&M.

Apsara Stack also provides full-stack support for a stable architecture to ensure the reliability of the cloud platform and business continuity.

System architecture of Apsara Stack

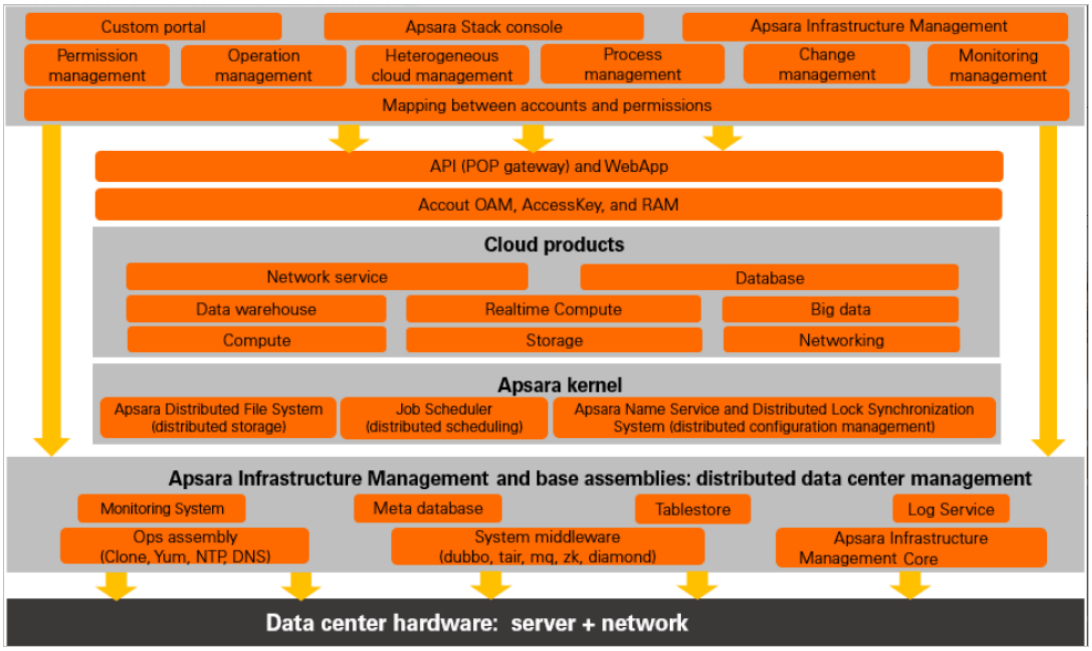



## Logical architecture

Apsara Stack virtualizes the computing and storage capabilities of physical servers and network devices to implement virtual computing, distributed storage, and software-defined networks on which ApsaraDB and big data services run. Apsara Stack provides underlying IT services for the applications of customers, and can be integrated with the existing account, monitoring, O&M systems of customers. The logical architecture of Apsara Stack has the following characteristics:

- The hardware infrastructure consists of servers and network devices. Only x86 servers are supported.
- The Apsara kernel of Apsara Distributed Operating System provides kernel services for all cloud services based on Apsara Distributed Operating System.
- All cloud services use the same API framework, security system, and O&M system. The O&M system is used for account management, authorization, monitoring, and logging.

Logical architecture of Apsara Stack




# 1.1.3.3. Network architecture

## 1.1.3.3.1. Overview

Apsara Stack adopts a flat Layer 2 Clos network architecture. The network architecture isolates the service plane from the out-of-band management plane, and supports the linear scaling and load sharing of switches.

As shown in Network architecture of Apsara Stack, the network architecture of Apsara Stack consists of four modules: internal network access module, Internet access module, data exchange module, and integrated access module.

- Internal network access module

  This module connects self-managed network resources to cloud resources and allows users to access virtual private clouds (VPC) or regular cloud services.

- Internet access module

  This module directly connects to the networks of Internet service providers (ISPs) or the backbone networks of customers. The business service area communicates with the Internet or other data centers by using this module. The business service area consists of the data exchange module and integrated access module.

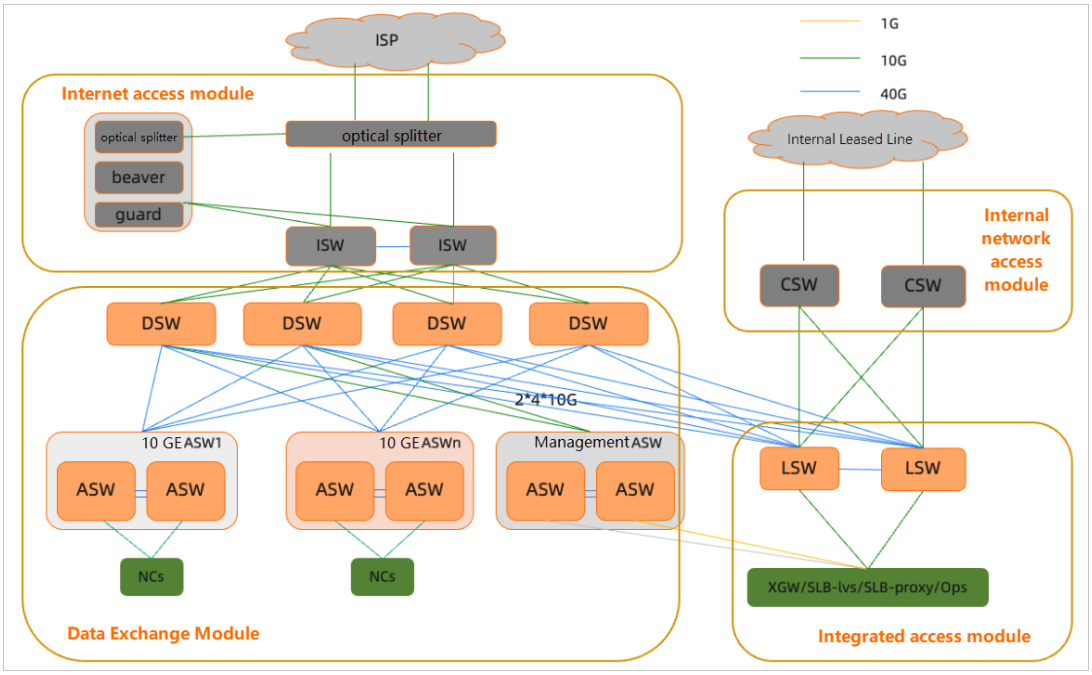
- Data exchange module

  This module provides access to all cloud service servers. The internal traffic among the servers is exchanged within this module.

- Integrated access module

  This module provides access to a variety of basic services and cloud services such as Server Load Balancer (SLB) and VPC.

Network architecture of Apsara Stack




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

| Role | Module | Description |
| --- | --- | --- |
| Inter-connection Switch (ISW) | Internet access module | An ISW is an egress switch and provides access to the networks of ISPs or the backbone networks of customers. |
| Customer Access Switch (CSW) | Internal network access module | A CSW facilitates access to the internal backbone networks of customers, including access to VPCs by using leased lines, and performs route distribution and interaction between the internal network and the Internet. |
| Distribution Switch (DSW) | Data exchange module | A DSW is a core switch that is connected to each Access Switch (ASW). |
| ASW | Data exchange module | An ASW provides access to cloud service servers and is uplinked with a DSW. |
| LVS Switch (LSW) | Integrated access module | An LSW provides access to cloud services, such as VPC and SLB. |

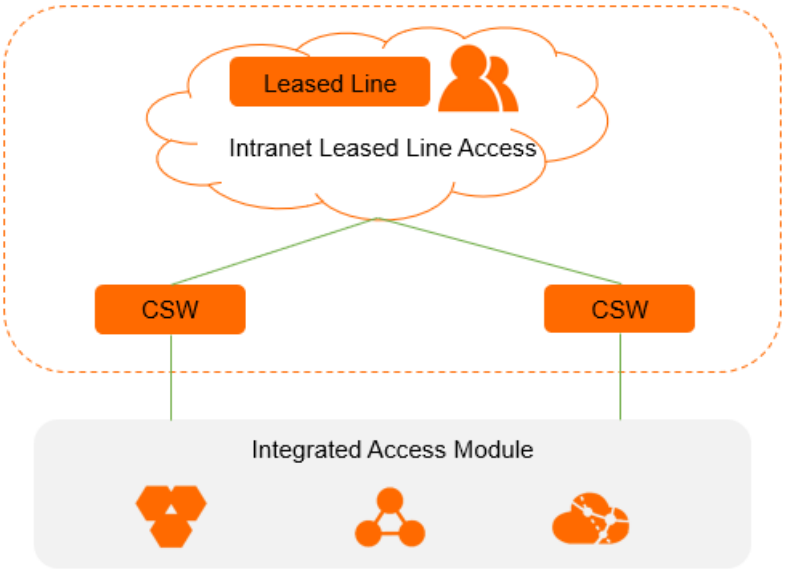
# 1.1.3.3.2. Internal network access module

This topic describes the components of the internal network access module.

Two CSWs provide internal users with access to VPCs and regular cloud services.

- Access to VPCs: The CSWs import internal users to VPCs by mapping internal users to VPCs. User groups on a CSW are isolated from each other.
- Access to regular cloud services: The CSWs are connected to the integrated access module by using External Border Gateway Protocol (EBGP) and allow direct access to all resources in the business service area.

Schematic diagram of the internal network access module




The leased line access solution of VPC allows customers to control their own virtual networks. For example, customers can select their own CIDR blocks and configure route tables and gateways. Customers 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 the smooth migration of applications to the cloud.

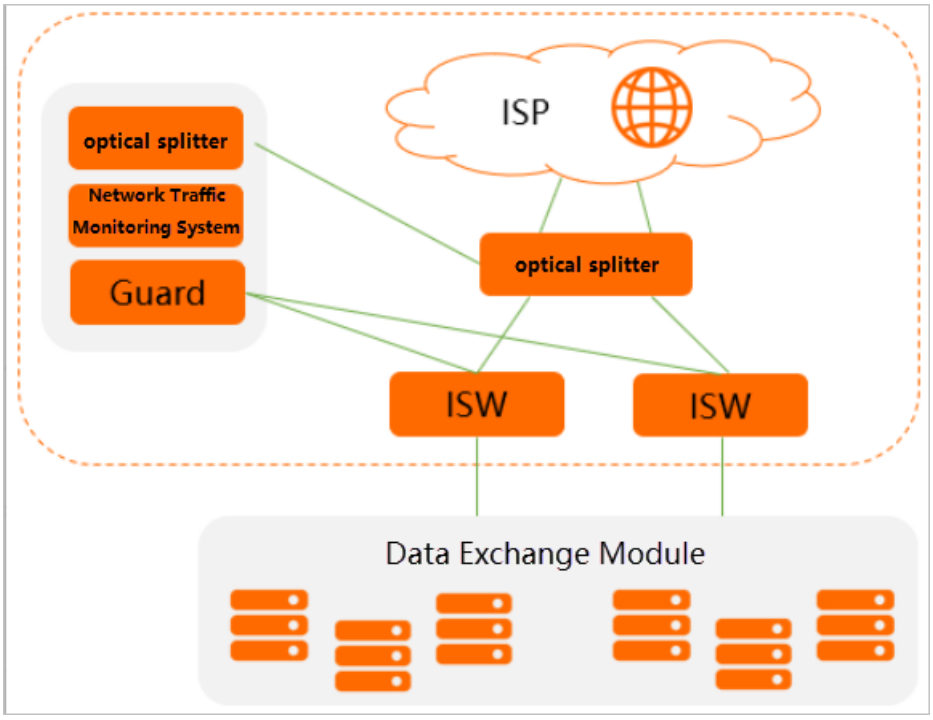
# 1.1.3.3.3. Internet access module

This topic describes the components of the Internet access module.

The Internet access module consists of two ISWs. This module facilitates access to the networks of ISPs or the public backbone networks of customers and performs route distribution and interaction between the internal network and the Internet. The two ISWs use Internal Border Gateway Protocol (IBGP) to back up routes between each other. The ISWs can use static routing or EBGP to uplink with the networks of ISPs or the public backbone networks of customers as required. The link bandwidth is determined by the Alibaba Cloud network size of customers and the bandwidth of their public backbone networks. We recommend that you connect the ISWs to the networks of multiple ISPs by using Border Gateway Protocol (BGP) to improve reliability. Each ISP can have 2 × 10 GE bandwidth. 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 from the data exchange module. This way, the interaction between the internal network and the Internet is implemented.

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, Network Traffic Monitoring System 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.

Schematic diagram of the Internet access module

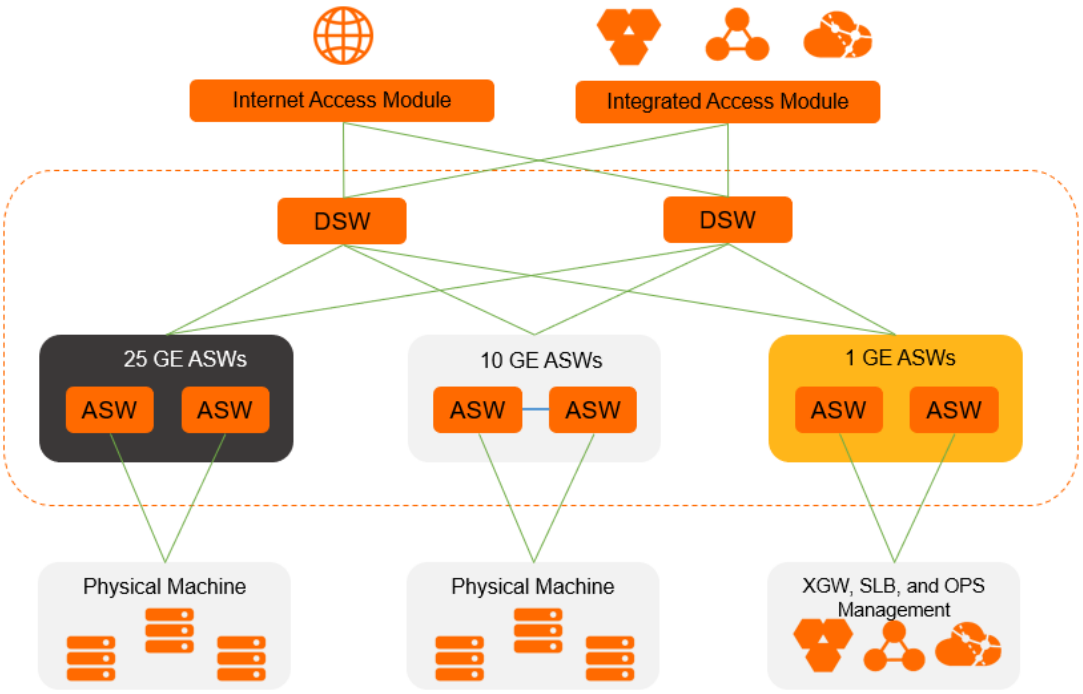



## 1.1.3.3.4. Data exchange module

This topic describes the architecture of the 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. The ASWs are connected to the DSWs by using 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.

Schematic diagram of the data exchange module

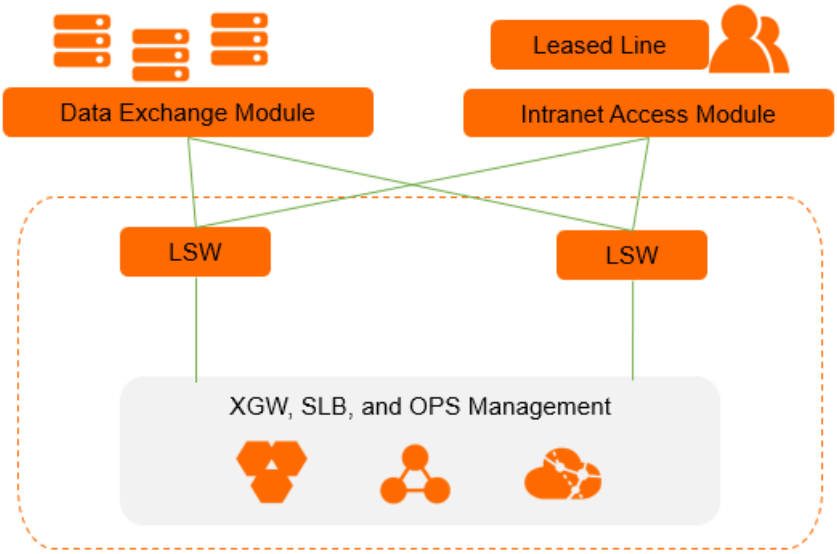



## 1.1.3.3.5. Integrated access module

This topic describes the components of the integrated access module.

Each cloud service server, which can be an XGW, SLB, or OPS server, 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 IBGP. The LSWs exchange routing information with DSWs and CSWs by using EBGP.

Schematic diagram of the integrated access module



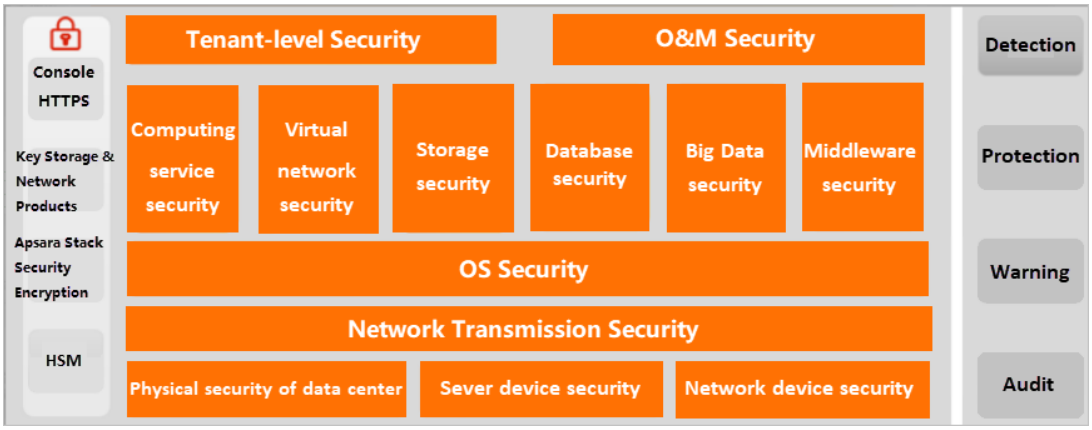

## 1.1.3.4. Security architecture

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

All service consoles on Apsara Stack must be accessed by using HTTPS. Apsara Stack provides a complete role-based authorization mechanism to ensure secure and controlled access to resources in multi-tenant mode. Apsara Stack supports a variety of security roles, including security administrators, system administrators, and security auditors.

In addition, Apsara Stack V3.0 and later use Alibaba Cloud Security to provide multi-level and integrated cloud security protection.

Hierarchical security architecture of Apsara Stack




# 1.1.4. Service panorama

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

## Hybrid cloud management services

Apsara Stack provides management services for hybrid clouds. The services allow you to elastically provide physical server resources to tenants on demand. This ensures that services such as core databases, key application systems, and high performance computing have computing resources as needed. The main service is Apsara Intelligent Operations Platform (ASIOP).

## IAAS services

Apsara Stack provides a wide variety of basic virtual resources, such as virtual computing, virtual networking, and virtual scheduling resources. The main services include Elastic Compute Service (ECS), VPC, Server Load Balancer (SLB), Container Service, Auto Scaling, Resource Orchestration Service (ROS), NAT Gateway, VPN Gateway, Elastic IP Address (EIP), Apsara Stack DNS, IPv6 Gateway, Container Registry, Express Connect, Cloud Gateway (CGW), Elastic High Performance Computing (E-HPC), and Server Migration Center (SMC).

Apsara Stack provides various storage services for different storage objects. The main services include Object Storage Service (OSS), Apsara File Storage NAS (NAS), Photo and Drive Service (PDS), Tablestore, and Cloud Defined Storage (CDS).

## Middleware services

Apsara Stack provides middleware services and can host various customer applications. This facilitates the conversion of applications to services and encourages applications to evolve into a microservices architecture. The main services include Message Queue for Apache RocketMQ, Message Queue for MQTT, Enterprise Distributed Application Service (EDAS), Application Real-Time Monitoring Service (ARMS), Message Queue for Apache Kafka, Application High Availability Service (AHAS), Global Transaction Service (GTS), Cloud Service Bus (CSB), Prometheus Service, and Multi-Site High Availability (MSHA).

### Application services

Apsara Stack provides API hosting, end-to-end logging, and cloud office services. The main services include API Gateway, Log Service, and Elastic Desktop Service (EDS).

### Database services

Apsara Stack provides a variety of database engines that can communicate with each other. The main services include ApsaraDB RDS, ApsaraDB for Redis, ApsaraDB for MongoDB, Data Management (DMS), AnalyticDB for PostgreSQL, PolarDB-X, Data Transmission Service (DTS), ApsaraDB for OceanBase, Data Lake Analytics (DLA), Graph Database (GDB), AnalyticDB for MySQL V3.0, AnalyticDB for MySQL V2.0, Database Autonomy Service (DAS), Database Backup (DBS), ApsaraDB for HBase, Time Series and Spatial-Temporal Database (TSDB), Advanced Database & Application Migration (ADAM), and ApsaraDB for Lindorm.

### Big data and AI 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, Elasticsearch, DataHub, Realtime Compute, DataQ, Apsara BigData Manager (ABM), E-MapReduce (EMR), Dataphin, Quick BI, Graph Analytics, Machine Learning Platform for AI (PAI), Hologres, and DataV.

### IoT services

Apsara Stack provides IoT services to deliver comprehensive cloud-based computing capabilities for a small amount of data. The main services include IoT Platform, Link IoT Edge, Security Operations Center, IoT Internet Device ID, and Link WAN.

### 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 main service is Apsara Stack Security.

### Apsara DevOps

Apsara Stack provides a one-stop DevOps platform that improves R&D efficiency through AI and automation technologies and enables the continuous delivery of values. The main services include Apsara DevOps, Apsara Hybrid Marketplace, and Sunfire.

## 1.1.5. 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

# 1.1.6. Compliance security solution

## 1.1.6.1. 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.

## 1.1.6.2. 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**

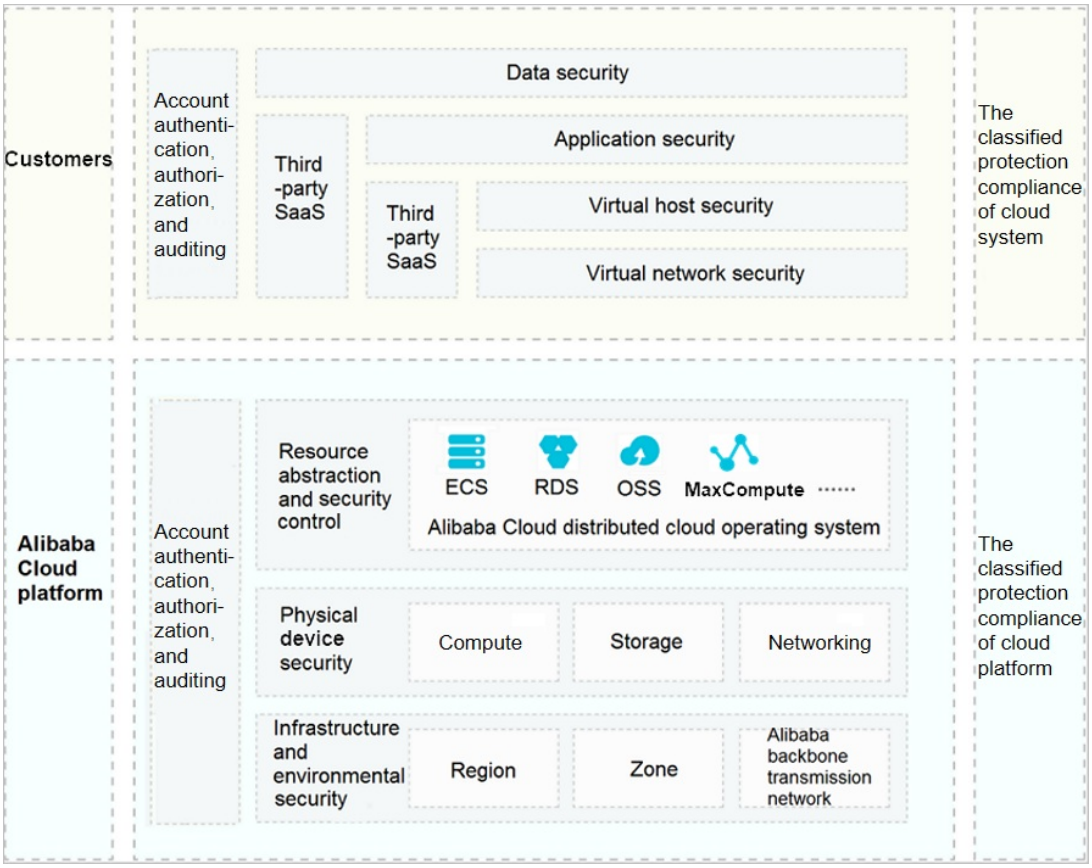
- The customer management staff can arrange, prepare, and implement the security policies and regulations as well as allocate management personnel based on the actual conditions of enterprises, and then formulate specialized documents.
- For the technical methods required in vulnerability management,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

# 1.1.6.3. 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

# 1.1.6.4. Implementation process of classified protection

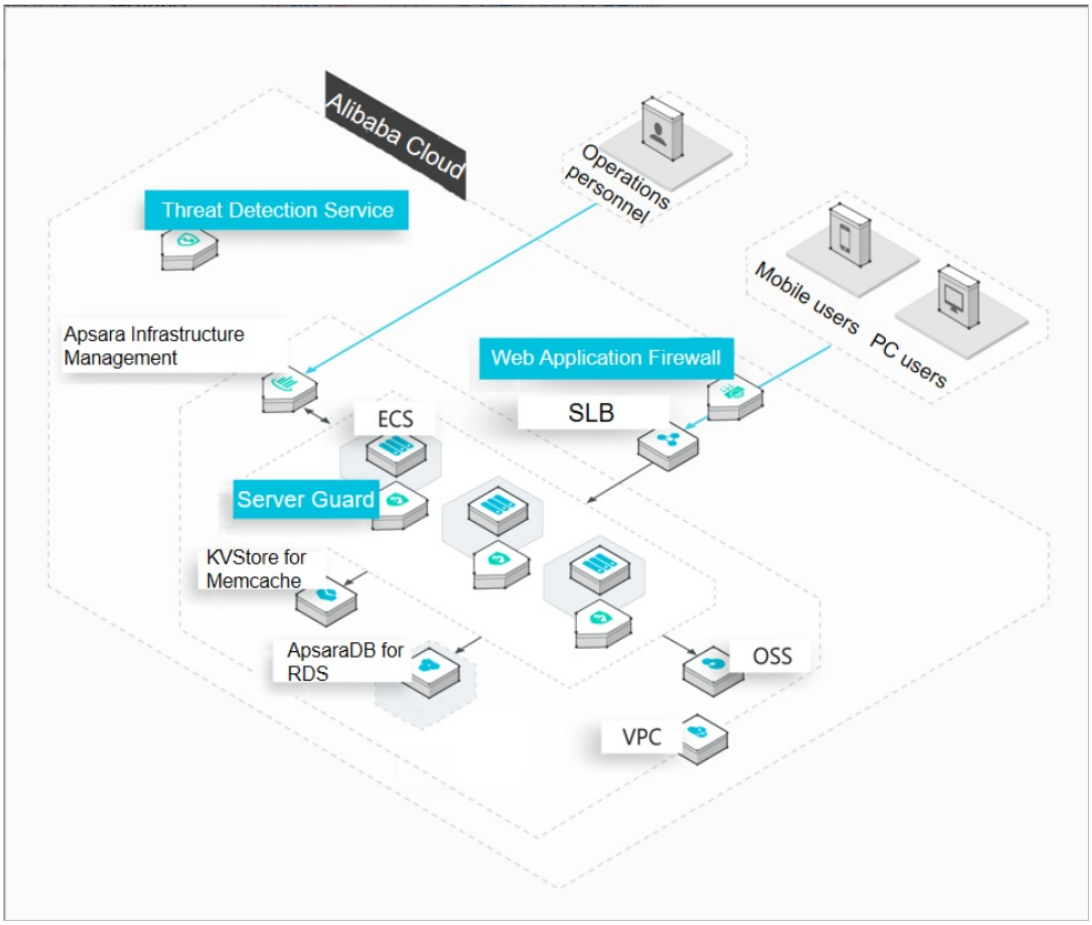
This topic describes the implementation process of classified protection. This topic also describes the responsible items of Alibaba Cloud and other entities that are involved in this process.

| | 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 |

# 1.1.6.5. Security compliance architecture

The security compliance architecture of Apsara Stack facilitates a fast connection to Apsara Stack Security and quick security rectifications. This allows Apsara Stack to comply with the basic technical requirements for classified protection at minimal security costs.

Security compliance architecture




Basic requirements for classified protection:

- Physical and environmental security: includes data center power supply, temperature and humidity control, windproof, rainproof, and lightning protection. 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.

# 1.1.6.6. Benefits

This topic describes the benefits of the Apsara Stack solution for compliance and security.

## One-stop assessment for classified protection

Alibaba Cloud works with high-performance consulting and assessment partners to provide one-stop compliance support throughout. This significantly saves costs for operation units.

- Eliminates multi-point communication and work redundancy to help operation units reduce investment.
- Improves efficiency by shortening the assessment cycle to as short as two weeks.

- Provides best practices of security and compliance on the cloud.

## Comprehensive security protection

The complete Apsara Stack Security architecture allows operation units to find corresponding services on Apsara Stack that can be used to rectify non-conformances. This helps operation units meet all requirements of classified protection.

# 2.Apsara Uni-manager

## 2.1. Product Introduction

### 2.1.1. 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, intelligent O&M, refined processes, and customizable scaling. Its simple management style brings excellent user experience that helps enterprises accelerate their digital transformation process.

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.

### 2.1.2. Benefits

This topic describes the benefits of the Apsara Uni-manager Management Console, Apsara Uni-manager Operations Console, and Apsara Uni-manager Dashboards, and their values to different user roles.

#### 2.1.2.1. Apsara Uni-manager Management Console

The Apsara Uni-manager Management Console simplifies the management and deployment of physical and virtual resources, helps build your own business systems in a simple and quick manner, improves resource utilization, and reduce operating costs.

##### 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 management from application perspectives.

##### Flexible permissions accelerate control

- Multiple preset roles for tasks such as operations management, resource usage, resource monitoring, and security management.
- Custom roles for flexible determination of shared permissions, managed resources, application

permissions, and menu permissions.

- Shared RAM authentication with public cloud and consistent permission management methods with public cloud.

### Intelligent analysis to expedite management

- Real-time update of global data and unified resource scheduling.
- Monitoring and analysis on resource usage trends, optimized resource distribution, and enhanced resource efficiency.
- Detailed metering and billing, visualized value of resources, and guaranteed service operations.

### 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.

## 2.1.2.2. Apsara Uni-manager Operations Console

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.

### Unified data monitoring

Unified monitoring on hybrid cloud resources, inventory, and alerts to check operating conditions, identify risks, and reduce and prevent accidents.

### Fast fault identification

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 identification 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

Shares inventory AI algorithms and dynamic analysis with public cloud to automatically calculate global optimal scaling policies, implement cost optimization, and reduce resource waste.

### Fast connection with third-party O&M systems

Provides visualized API portals, standard northbound interfaces, and page-level integration settings for fast connection with third-party O&M systems.

# 2.1.2.3. Apsara Uni-manager Dashboards

## Diverse business scenarios

Standard preset dashboards are provided and custom dashboards are supported to cater for diverse business scenarios.

## Various data sources

Various data sources of JSON, ASAPI, ASAPI data pool, HTTP, JSONP, and Excel are supported for comprehensive O&M data.

## Flexible custom dashboards

Rich visualization components and data sources allow you to customize different dashboards.

## Fast data retrieval

Data presentations in bar charts, pie charts, dashboards, and maps help you quickly obtain information and improve O&M efficiency.

# 2.1.2.4. Benefits to different user roles

Apsara Uni-manager brings value to decision makers, operations administrators, ordinary users, and O&M engineers, and improves user experience.

| Role | Focus | Benefit |
| --- | --- | --- |
| Decision-maker | Resource presentation | Panoramic resource presentation capabilities, including:<br>• Panoramic data visualization<br>• Regular generation of resource reports<br>• Intelligent recommendations on the homepage<br>• Resource presentation on multiple terminals such as large screens and mobile terminals<br>• Custom analysis presentation capabilities |
| Operations administrator | Resource operations | Complete hybrid cloud and multi-level cloud operations capabilities, including:<br>• Centralized management of resources and permissions<br>• Automated processing and product upgrades<br>• Forecasts and optimization suggestions for resource quotas<br>• Expansion of operations capabilities such as metering and billing |

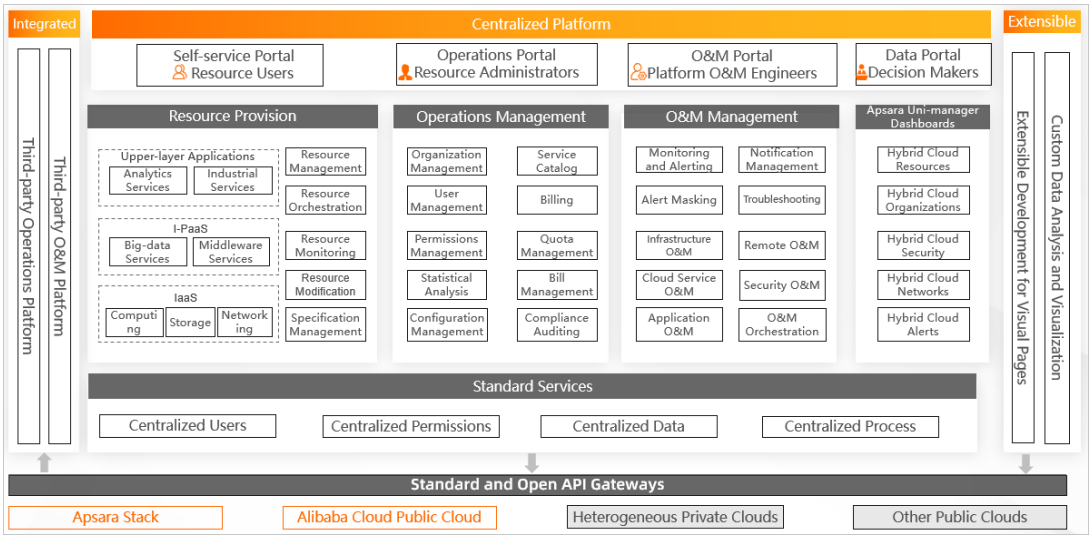
| Role | Focus | Benefit |
| --- | --- | --- |
| Ordinary user | Resource supply | Comprehensive and flexible resource supply capabilities, including:<br>• Centralized activation and management of multi-level cloud resources<br>• Automatic orchestration and elastic scaling of cloud resources<br>• Suggestions on resource allocation and cost optimization<br>• Flexible connection with and use of third-party cloud products<br>• 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:<br>• Centralized presentation of various types of monitoring data<br>• Automated O&M scripts and tools<br>• Root cause analysis and locating of faults<br>• Fast connection with third-party O&M systems |

# 2.1.3. Architecture

This topic describes the architecture of Apsara Uni-manager.

The following figure shows the architecture of Apsara Uni-manager.

Apsara Uni-manager architecture




## Unified portals

Apsara Uni-manager provides unified portals: self-service portal, management portal, operations portal, and data dashboard portal. They allow different users to manage cloud resources.

- **Self-service portal**: provides abilities for ordinary users to autonomously manage all cloud products and cloud resources, such as resource monitoring and resource operations.
  - IaaS: provides IaaS layer services such as computing, storage, and networking to build stable and complete underlying resource architectures.
  - I-PaaS: provides PaaS layer services such as middleware and big data to deliver high-performance support for building agile cloud-native applications.
  - Upper-layer applications: Services such as QuickBI and Machine Learning Platform for AI are provided to help quickly build upper-layer applications.
  - Resource management: provides Resource Management capabilities including creation, deletion, and configuration to meet all-round and multi-scenario resource control requirements.
  - Resource orchestration: You can use resource orchestration templates to automate the deployment of multiple types of resources. This improves resource creation efficiency and standardizes resource creation processes.
  - Resource monitoring: You can specify alert rules to monitor and alert resource usage. This ensures the stability of resources.
  - Resource changes: You can change the organizations and resource sets for resources to meet the requirements of migrating resources due to organizational structure adjustment.
  - Specification management: allows you to control specifications when you create resources for flexible resource provisioning.
- **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.
  - Organization management: allows you to manage the lifecycle of organizations to satisfy the requirements for enterprise organizations. You can create and deleting organizations and add users to organizations.
  - User management: allows you to manage the lifecycle of users. You can create, delete, and modify users.
  - Permission management: You can create roles, authorize roles, and control permissions, so that users have different capabilities to view and manage resources.
  - Compliance and auditing: provides operation logs for security auditing.
  - Quota management: You can set quotas on different resources for organizations and resource sets and control the total resources used by organizations to improve the efficiency of resource allocation and usage.
  - Metering and billing: meters resource by organization and resource set, and formulates flexible billing rules.
  - Statistical analysis: analyzes resource usage and costs for organizations and resource sets.
  - Service catalogs: You can customize service catalogs and control available service catalogs.
  - Bill management: allows you to manage bills in different dimensions, such as organizations, resource sets, and resources.

  - Configuration management: provides configuration capabilities such as security policies, personalized configurations, and menu management to cater for different scenarios.

- **Operations portal**: provides O&M engineers with integrated O&M capabilities, such as monitoring and alerting, fault locating, operation changes, and security O&M.
  - Monitoring and alerting: aggregates and displays monitoring and alerting data of products. This allows you to check system conditions and identify hidden risks.
  - Notification management: sends notifications of alerts by email and DingTalk.
  - Alert blocking: During maintenance and upgrade windows, you can set alert blocking filters in multiple dimensions such as product, cluster, machine, service, and alerted item.
  - Fault locating: provides multi-level displays of product architecture, dependencies, and deployment information. After a fault occurs, the root causes can be analyzed to determine its impacts.
  - Infrastructure O&M: includes the physical server O&M and network infrastructure O&M.
    - Physical server O&M: displays the device information and monitoring information of physical servers in a unified manner. The data center information of physical servers can also be provided.
    - Network infrastructure O&M: aggregates network device information, monitoring information, and cluster usage. Physical network topology can be displayed and analyzed.
  - Remote O&M: provides professional, secure, and cost-effective managed remote O&M services.
  - Product O&M: involves information such as products, clusters, server roles, machines, alerts, final state or not. You can view information in a drill-down manner: products, clusters, and server roles.
  - Security O&M: allows you to audit CLI operations and block high-risk actions. This overcomes the traceability and auditing difficulties with CLI operations.
  - Application O&M: You can integrate the application O&M module to display application O&M views.
    - The full application topology allows you to discover faulty applications and view the upstream and downstream application information of applications.
    - The overview of an application displays the called upstream and downstream applications and the health of the inbound and outbound traffic for the current application. This can visually present the affected business of the current application.
  - Orchestration O&M: You can use built-in public templates or create custom templates to orchestrate O&M actions for automated O&M.

- **Data dashboard portal**: provides decision-makers with multidimensional and panoramic data presentation capabilities. Five dashboards are provisioned: resources, organizations, security, network, and alerts. Personalized capability expansion is supported.
  - Resources dashboard: provides an overview of resources for multiple regions and multiple clouds. You can view the resource usage in the region dimension.
  - Organizations dashboard: provides an overview of resources for multiple organizations. You can view the resource usage in the organization dimension.

- Security dashboard: provides an overview of security information such as security protection, vulnerability detection, and intrusion detection. You can view protection levels in the security dimension.
- Network dashboard: provides an overview of physical network information. You can view network usage in the network dimension.
- Alerts dashboard: provides an overview of various types of alerts.

### Unified services

Apsara Uni-manager provides unified service capabilities, including unified users, unified permissions, unified data, and unified process management.

### Openness, ease of integration, and scalability

Apsara Uni-manager furnishes multi-cloud control capabilities and open API gateways. Northbound API gateways are provided to third-party integrators to integrate third-party data and pages in the form of data collection. Southbound API gateways are used for integration with and adaptation to multi-cloud environments.

## 2.1.4. Features

This topic describes the features of the Apsara Uni-manager Management Console, Apsara Uni-manager Operations Console, and Apsara Uni-manager Dashboards.

## 2.1.4.1. Apsara Uni-manager Management Console

The Apsara Uni-manager Management Console provides features such as resource management, personnel management, permission management, operations center, security center. This simplifies the management and deployment of physical and virtual resources, enhances resource usage, and reduces operating costs.

### Resource management

You can activate and monitor resources. This helps you understand the usage of each resource and avoid risks in a timely manner when exceptions occur.

| Feature | Description |
| --- | --- |
| Products | You can activate more than 80 services and resources that fall into categories such as computing, storage, networking, database, big data, middleware, application, and security. |

| Feature | Description |
| --- | --- |
| Resource sets | • You can create resource sets with custom names. Resource set names can be modified multiple times.<br>• You can obtain resource set information by organization or resource set name. The resource list and member list are displayed for the current resource set.<br>• You can add members to or remove members from the current resource set. Members can be users or user groups. After a member is removed, it has no permissions to access the resource set.<br>• You can delete non-default resource sets that contain no resources or members. |
| Home dashboards | You can modify modules and homepage layout. You can select quota overview, total number of instances, historical number of resources, resource loads, alerts, my management, and regional distribution, and then display them in charts. You can drag modules to adjust the homepage layout. |
| CloudMonitor | • You can view the monitoring charts and alert rules for an instance of each service.<br>• You can create alert rules for the current instance.<br>• You can view the historical alert list, alert rule list, and alert rule details.<br>• You can set the thresholds of metrics in alert rules. Alert notifications are sent when alert conditions for metrics are triggered.<br>• You can enable, disable, modify, and delete alert rules. |

## Personnel management

**Organization management, user management, and user group management allow enterprises to control the organizations, permissions, and groups of users in a centralized manner. This can improve management efficiency by administering the full life cycle of users in different scenarios and satisfying the access needs of different users to systems and resources.**

| Feature | Description |
| --- | --- |
| Organization management | • You can drag organization names in the organization list to customize the organizational hierarchy.<br>• You can create organizations with custom names. Organization names can be modified multiple times.<br>• You can view the organization information. The resource set list, user list, and user group list are displayed for the current organization.<br>• You can change the ownership of an organization and delete an organization.<br>• You can update the association between an organization and a region. |

| Feature | Description |
| --- | --- |
| User management | • You can view the system user list and historical user list.<br>• You can create and delete users. You can select the organization, role, and logon policy for a new user. You can assign multiple roles to a user.<br>• You can modify the user information, user role, user group, and logon policy of the current user.<br>• Allows you to view the initial password of a user and reset the password.<br>• You can disable and enable system users. By default, new users are enabled.<br>• You can restore historical users. |
| User group management | • You can create and delete user groups. You can select the organization and role for a new user group. You can assign multiple roles to a user group.<br>• You can modify the role of a user group.<br>• You can add users to or remove users from a user group.<br>• You can modify user group names. |
| User center | • You can view your personal information and notification methods.<br>• You can modify your personal information, logon password, and notification methods.<br>• You can switch the current role and set the new role as default.<br>• You can view the policy information and RAM roles of the current role. The policy information is displayed in the JSON format. |

## Permission management

**You can configure roles, RAM roles, data permissions, and access to the system and services to improve system security.**

| Feature | Description |
| --- | --- |
| Role management | • You can view role information. Management permissions, application permissions, menu permissions, and authorized personnel are displayed for roles.<br>• You can create custom roles and select permissions for custom roles.<br>• You can modify the basic information and permissions of custom roles.<br>• You can disable and enable roles. By default, new roles are enabled.<br>• You can duplicate roles and modify permissions of duplicated roles.<br>• You can delete custom roles that are not bound to user groups. |

| Feature | Description |
| --- | --- |
| Data permission management | • You can grant data permissions to Message Queue instances, OSS instances, Log Service instances, DataHub instances, Container Service for Kubernetes instances. You can enable or disable the permissions based on your business requirements.<br>• You can view the user list in the current organization and user permissions. The permission information is displayed in the JSON format.<br>• You can modify the permissions of the current user in the JSON format. |
| Access management | • You can view access policy information.<br>• You can create and modify access policies. You cannot modify default policies.<br>• You can enable or disable access policies. You cannot enable or disable default policies.<br>• You can delete non-default policies that are not bound to users. |
| RAM role management | You can manage and configure RAM roles in the RAM console to control user access to resources in a centralized manner.<br>• You can view the basic information of RAM roles.<br>• You can create RAM roles and select the sharing scope, existing permission policies, and user groups for RAM roles.<br>• You can add, modify, and delete permission policies for RAM roles. Policy information must be in the JSON format.<br>• You can view the details of RAM roles. |
| Service-linked role management | • You can view the basic information and details of service-linked roles.<br>• You can create service-linked roles and select organization names and service names for service-linked roles.<br>• You can view permission policies for service-linked roles. Policy information is displayed in the JSON format. |

## Operations center

**Quota management, metering and billing management, statistical analysis, and bill management allow enterprises to control resource usage and resource billing in a centralized way. This helps you quickly understand resource usage and billing information, and provides flexible control features.**

| Feature | Description |
| --- | --- |
| Quota management | • You can view the quotas of services in resource sets of your organization.<br>• You can create, modify, and clear quotas of services.<br>• You can create quota alerts and specify alert objects and alert information for quota alerts.<br>• You can delete quota alerts.<br>• You can view the quota alert history. |

| Feature | Description |
| --- | --- |
| Metering and billing management | • You can view statistics about the number of resource instances that run in the Apsara Stack environment by time, organization, resource set, or region. You can also export statistical reports.<br>• You can view the numbers of billing plans, policies, and rules that are available and in use. You can also view the effective time of billing plans and policies.<br>• Billing rules:<br>  ○ You can view the list of billing rules. You can also view the details of a specified billing rule.<br>  ○ You can view, create, modify, clone, and delete billing item. A billing rule must contain one or more billing items.<br>  ○ You can create, modify, clone, and delete billing rules. You cannot delete default billing rules.<br>• Billing policies:<br>  ○ You can view the list of billing policies. You can also view the details of a specified billing policy.<br>  ○ You can create, modify, clone, and delete billing policies. You cannot delete default billing policies.<br>• Billing plans:<br>  ○ You can view the list of billing plans. You can also view the details of a specified billing plan.<br>  ○ You can create, modify, clone, and delete billing plans. You cannot delete default billing plans. |
| Statistical analysis | • You can view resource reports, quota reports, and CloudMonitor reports.<br>• You can export full reports or create report export tasks.<br>• You can view existing report export tasks.<br>• You can download or delete selected reports.<br>• You can create download tasks to export reports of services in the specified time range, resource set, and region. |
| Tag management | • You can create tags, add tags to resources, and search for tags.<br>• You can manage tags in a centralized manner.<br>• You can view bound resources and information of tags. |

| Feature | Description |
| --- | --- |
| Process management | • Approval forms:<br>The resource approval form displays the requests submitted by the current user and pending approval. The resource approval process is completed through approval.<br>• Process configuration:<br>You can manage process configurations. You can configure approval processes for different organizations and services. You can configure resource application processes only for some services. This feature will cover more services and more scenarios.<br>• Process management:<br>You can create process definitions and define approval processes for different organizations and services. |
| Bill management | • Service bills:<br>◦ You can view the bill overview of services. Consumption trends of services, consumption distribution of top 10 services, and consumption details of all services in the past six months are displayed in charts.<br>◦ You can view and export bill statistics and bill details of services.<br>• Organization and resource set bills:<br>◦ You can view the bill overview of organizations and resource sets. Consumption trends of organizations, consumption distribution of top 10 resource sets, and consumption details of organizations in the past six months are displayed in charts.<br>◦ You can view and export bill statistics and bill details of organizations and resource sets.<br>• Billing details: You can view and export billing details by organization and resource set. |

## Security center

**Operation logs, multi-factor authentication (MFA), and AccessKey pairs allow you to reinforce security for system access and resources.**

| Feature | Description |
| --- | --- |
| Operation logs | • You can view operations logs and select filters for operation logs.<br>• You can export the logs displayed on the current page and save them to your computer as a CSV file. |
| MFA | • You can check the MFA feature is enabled. If not, you can enable it. However, you cannot disable it after you enable it.<br>• You can bind or unbind virtual MFA devices.<br>• You can reset MFA keys. |

| Feature | Description |
| --- | --- |
| AccessKey pairs | • You can view your AccessKey pair.<br>• You can create AccessKey pairs. You can have at most two AccessKey pairs.<br>• You can enable and disable AccessKey pairs. Make sure that at least one of the AccessKey pairs is enabled. By default, a new AccessKey is enabled after it is created.<br>• You can delete AccessKey pairs. At least one of the AccessKey pairs must be retained. |

# 2.1.4.2. Apsara Uni-manager Operations Console

The Apsara Uni-manager Operations Console provides a unified portal for features such as general O&M, product O&M, security & compliance, and system settings. This simplifies routine O&M and improves O&M efficiency.

## Homepage

The homepage allows you to view the statistics and summary data of alerts, physical resources, and service inventory.

The following table lists the information on the homepage.

| Section | Description |
| --- | --- |
| Username | The user and its department. |
| Alert Overview | The number of regions without alerts, the number and details of regions with critical alerts, the total number and details of critical alerts. The distribution of regions without alerts and regions with critical alerts is also displayed in the map. |
| Resource Overview | The total numbers of racks, servers, and network devices. |
| Quotas and Usage | The quotas and usage of ECS, SLB, RDS, OSS, Tablestore, Log Service, and NAS. |

## General O&M

General O&M covers alert management, inspection management, resource management, capacity management, change management, and backup management.

The following table lists the main features of general O&M.

| Feature | Description |
| --- | --- |

| Feature | Description |
| --- | --- |
| Alert management | • Alert overview: You can view alerts by region and service, and alert details.<br>• Alert list: You can view and handle all alerts in the current region. You can filter alerts based on your needs and export alerts in the Excel format.<br>• Alert settings:<br>  ○ You can set alert contacts and static parameters for timeout alerts.<br>  ○ System alert templates are provided. You can customize alert templates to modify alert trigger rules based on your needs.<br>  ○ You can set alert notification channels and alert push parameters. Alert notifications can be pushed by DingTalk and email.<br>  ○ You can add and remove alert blocking rules.<br>• Alert packages: You can upload alert packages for hot replacement of alert data. |
| Inspection management | • One-click inspection: You can start preset inspection tasks and custom inspection tasks. You can view recent inspection results.<br>• Inspection dashboard: displays recent inspection tasks, data overview, distribution and trends of inspection exceptions, inspection task records, inspection issues, and latest inspection reports.<br>• Inspection reports: You can view all inspection reports to check issues or faults of the system.<br>• Inspection scenarios: You can view, add, modify, and delete inspection scenarios.<br>• Inspection records: You can query system inspection records and related inspection reports. You can stop ongoing inspection tasks.<br>• Inspection items: You can view the details of all inspection items.<br>• Inspection packages: You can import and export inspection packages. |

| Feature | Description |
| --- | --- |
| Resource management | • Products:<br>  ○ You can view all resources in the product dimension, including the details of products, clusters, services, server roles, and virtual machines.<br>  ○ You can restart a specified server role.<br>  ○ You can perform security O&M on a specified server role.<br>• Data centers:<br>  ○ You can view information about cabinets, servers, and server roles in data centers.<br>  ○ You can restart a specified server role.<br>  ○ You can perform security O&M on a specified server role.<br>  ○ You can view the monitoring information of physical machines, and view, fix, and delete alerts.<br>• Network:<br>  ○ You can view the standard topology of all network devices on the platform.<br>  ○ You can view the real-time topology of all network devices on the platform.<br>• Resource tags:<br>  ○ You can add, view, and delete the resources that you follow.<br>  ○ You can add, view, bind, delete, and export resource tags. |
| Capacity management | • Capacity analysis: You can view the average available inventory and changing trends of services, and the usage of core services.<br>• ECS: You can view and export ECS inventory details, view CPU and memory details of all ECS instance families, and set usage thresholds.<br>• SLB: You can view and export SLB usage details, view the usage of internal VIPs and public VIPs, view network interface controller traffic, and set usage thresholds.<br>• OSS: You can view and export OSS inventory details, view available and used OSS inventory, and set usage thresholds.<br>• Tablestore: You can view and export Tablestore inventory details, view available and used Tablestore inventory, and set usage thresholds.<br>• Log Service:<br>  ○ You can view and export Log Service inventory details, view historical inventory records and current quota details of base Log Service, and set the current quota threshold and global quota.<br>  ○ You can view the available and used inventory of applied Log Service, and set the current quota threshold and global quota.<br>• EBS: You can view available and used EBS inventory, and inventory details.<br>• NAS: You can view available and used NAS inventory, and inventory details. You can also export NAS inventory details.<br>• RDS: You can view and export RDS inventory details, view recent RDS inventory history, and set usage thresholds. |
| | • Operation Orchestration Service: |

| Feature | Description |
| --- | --- |
| Change management | ○ You can view information about host resources (including physical machines and Docker virtual machines), such as the hostname, IP address, project name, cluster name, operating system, and IDC.<br>○ You can view information about Docker resources, such as the server role name, type, hostname, host IP address, project name, cluster name, and service name.<br>○ Common default script libraries are provided and custom scripts are supported. You can view, modify, import, and export scripts.<br>Script libraries store scripts that implement various features. Scripts can be written in the Python and Shell formats.<br>○ You can customize O&M jobs. You can view, modify, execute, import, export, and delete O&M jobs.<br>Each O&M job is a collection of O&M resources, software, and scripts. Scripts are used to implement features and are executed on different hosts or Docker instances in a specified order.<br>○ You can customize process orchestration. You can view, modify, execute, import, export, and delete review processes. The jobs and overall process of process orchestration can be displayed in charts.<br>Process orchestration combines a series of logical actions (including scripts and jobs) into an automated O&M task.<br>○ You can create, import, view, export, modify, execute, and delete O&M jobs.<br>○ You can view and delete the execution history of O&M jobs. You can resume O&M jobs that support phased execution. You can use snapshots to view the list of transfer files, execution scripts, and execution hosts contained historical O&M jobs.<br>○ You can view execution results, and approve or stop O&M jobs.<br>○ You can view execution results, and approve or stop processes.<br>○ You can view logs of various automated O&M tasks.<br>• Log cleanup:<br>○ You can view the usage information of containers and physical machines.<br>○ You can manually and automatically clear log files from specified containers (Dockers) or physical machines (virtual machines or bare metal machines).<br>○ You can import, export, modify, and delete log cleanup rules, and view log cleanup records.<br>• Security O&M<br>○ You can implement fast logon to devices in the console, and upload and download files.<br>The following devices support fast logon:<br>■ The virtual machines, hosts, and containers where server roles reside<br>■ The metadatabases used by server roles<br>○ You can view the environment metadata, OOB information, and cluster configurations for the current user.<br>○ You can audit command records, file upload and download records, authorization records, and command videos.<br>○ You can view, modify, import, export, and delete Linux command rules. The rules can be used to block, approve, confirm, and verify Linux commands.<br>○ You can configure workers and whitelists to connect to Apsara Stack Online for remote O&M over secure network channels. |

| Feature | Description |
| --- | --- |
| Backup management | You can add backup products and configure backup items and backup servers to perform offline backup of metadata for Apsara Distributed File System and OPS-DNS. You can view details of offline backup tasks. |

## Product O&M

**Product O&M involves products in the following categories: computing, networking, storage, database, middleware, big data, platform, security, and application. Compute Operations Console, Network Service Diagnosis, Network Operations Console, and Apsara Distributed File System are integrated into the Apsara Uni-manager Operations Console. For other product, you are redirected to their O&M consoles for fine-grained O&M.**

**The following table lists the main features of product O&M.**

| Feature | Description |
| --- | --- |
| Compute Operations Console | • You can obtain the core metrics of ECS, check ECS status, view overview information such as product detection, alerts, inventory usage, key metrics, and health status. One-click search is supported.<br>• You can view information of all computing clusters, log on to cluster AGs, view computing cluster overview, configure computing clusters, view computing server information, and change the status of computing servers.<br>• You can view the basic information and O&M details of computing servers. You can diagnose machines, lock and unlock NCs, launch features, and perform overall migration. You can view the audit logs of the preceding operations and the history of migration tasks.<br>• You can implement ECS O&M.<br>  ◦ You can view the ECS instance list and their details, such as the basic information, configurations, performance monitoring, and associated resources. You can diagnose ECS instances, migrate ECS instances, view migration history of ECS instances, change instance status, log on to VNC, manage ISO files, and view the audit logs of the preceding operations.<br>  ◦ You can view the disk list and their details, detach disks, create and view snapshots, and view the audit logs of the preceding operations.<br>  ◦ You can view the list of images and the status of ISO files that can be mounted.<br>  ◦ You can view and delete snapshots, create images from snapshots, view the audit logs of the preceding operations, and view automatic snapshot policies.<br>  ◦ You can view the ENI list and their details, unbind and release ENIs, and view the audit logs of the preceding operations.<br>  ◦ You can view the security group list and their details, and the audit logs related to security groups.<br>  ◦ You can view, add, modify, and delete custom instance types. You can view the audit logs of the preceding operations.<br>• You can view audit logs of all operations.<br>• You can view the monitoring information of servers, including databases, scheduled tasks, abnormal workflows, and workflow queues.<br>• You can query the inventory of ECS instance types to check their usage trends. |

| Feature | Description |
| --- | --- |
| Network Service Diagnosis | • You can view the diagnostic information of historical instances.<br>• You can diagnose SLB instances and DNS instances based on your business requirements.<br>• You can view the details of historical intelligent path analysis tasks, and create and terminate analysis tasks. |
| Network Operations Console | Network Operations Console provides operations capabilities such as the visualization of network-wide monitoring, automated implementation, automated fault location, and network traffic analysis to enhance the efficiency of network operations engineers and reduce operations risks.<br>• Dashboard: You can view the basic information, running status, and traffic monitoring of network devices. You can obtain network topology and link information. You can customize network views to display network monitoring data.<br>• Network element management: You can view the basic information, running status, traffic monitoring, and alerts of network devices. You can view and modify the collection cycle of device information, and add and modify OOB CIDR blocks. You can modify passwords of network devices. You can compare consistency between the current configurations of a network and its startup configurations.<br>• SLB cluster management: You can set tags for SLB clusters.<br>• SLB management: You can view the basic information, node information, and usage data of SLB in the cluster monitoring and instance monitoring dimensions.<br>• SLB proxy management: You can view the node status, usage data, and aggregate monitoring metrics of SLB proxy clusters in the cluster monitoring dimension. You can also view the usage chart of metrics in the instance monitoring dimension.<br>• Anytunnel management: You can view the registration information of anytunnel resources. The LB_ID, VIP, project, cluster, instance, and server role filters are supported.<br>• XGW management: You can view the usage data of VPC gateway nodes and VPC gateway instances.<br>• CGW management: You can view the usage data of CGW nodes and CGW instances.<br>• Cloud firewall management: Yon can configure bypass isolation and isolation & restoration for firewalls.<br>• Alert management: You can add traps to report alerts when monitoring metrics for network devices are abnormal. You can view and handle live alerts. You can view traps and historical alerts.<br>• IP address conflicts: You can check whether conflicting IP addresses exist in the current Apsara Stack environment.<br>• Leased line detection: You can configure parameters, generate configurations, send configurations to specified devices, and perform leased line detection tests in the console.<br>• Baseline configuration audit: You can view audit results for comparison between baseline configurations and current configurations of network devices.<br>• Network inspections: You can create, modify, and delete inspection templates, although preset inspection templates are provided. You can create one-time or scheduled inspection tasks based on templates. You can view, modify, start, suspend, and delete inspection tasks. You can view the inspection data for the day and the last 10 inspection records in the inspection dashboard. You can view inspection items and inspection history. |

| Feature | Description |
| --- | --- |
| | • Hybrid cloud network: You can view cross-cloud access records. You can configure dynamic VIPs and dynamic DNS records to implement connections between services in multiple clouds and between services in IDCs and Apsara Stack. You can manage data for network access of cloud services in hybrid clouds.<br>• Cloud access gateway O&M: You can view the summary information of cloud access gateway instances, the network information of bare metal machines in VPCs, and the history of bare metal API operations. You can perform the following operations related to Cloud access gateways: check HSW initialization configurations, bare metal routing configurations, and bare metal network gateway information, apply for and release bare metal machines in VPCs, delete VPC route table entries, VBR route table entries, VPC router interfaces, VBR router interfaces, VBRs, and physical connections, clear resources with one click, view and modify physical connection broadband, view trunk usage, and view BM VPN and BD usage.<br>• Network security and protection: You can view and modify border protection policies, and carry out SRS O&M and Donghuangzhong O&M.<br>• Hybrid cloud resources: You can view the physical network topologies of hybrid clouds from multiple perspectives. You can manage Apsara Stack data centers, user-managed data centers, and their network element devices. You can view the IP address pools applied for different products in instances. |
| Apsara Distributed File System | • You can view the following information of Apsara Distributed File System clusters: O&M status, overview, alert monitoring data, replica information, cluster trend chart, rack information, master information, and CS information.<br>• You can view the following information of services: storage space, server information, health information, health heatmap, and top 5 abnormal services.<br>• You can modify preset alert thresholds for clusters. You can go to the Apsara Distributed File System Portal page with one click. |
| EBS | • You can view NC information, virtual machine information, and block device information.<br>• You can view the overview and trend charts of EBS clusters in the dashboard.<br>• You can view information such as the address status of block master nodes, block server nodes, snapshot server nodes, and block Gcworker nodes in EBS clusters. You can also switch leaders for nodes and query and configure flags.<br>• You can view information such as the ID, status, capacity, and type of disks for EBS clusters. You can modify disk configurations. You can flush disks or segment transaction logs on disks. You can enable, disable, delete, and restore disks. You can view and redistribute segments.<br>• You can view the affected virtual machine (VM) list, VM cluster statistics, and device cluster statistics on the IO HANG page.<br>• You can view the slow IO list, top ten NCs, cluster statistics, top five cluster statistics, and reasons on the SLOW IO page.<br>• You can view the sales status of a cluster, configure the oversold ratio of a cluster, and specify whether a cluster is on sale on the Product Settings page. |

## Security and compliance

Security and compliance involves operation log audit, server password management, AccessKey pair management, and platform encryption management.

The following table lists the main features of security and compliance.

| Feature | Description |
| --- | --- |
| Operation log audit | You can view logs to check the resource usage and running status of all modules on the platform. |
| AccessKey pair management | This feature is hidden by default. You can view base AccessKey pairs and create AccessKey pair rotation tasks. This feature significantly affects the base. You must contact Alibaba Cloud technical support. |
| Server password management | You can view server passwords and their information, update passwords one by one or in batches, view and modify password policies, and view update records and historical passwords. |
| Platform encryption management | • You can perform disk encryption for metadatabases by using two encryption methods: SM4 encryption algorithm and AES encryption algorithm. You can view disk encryption history.<br>• You can perform transmission encryption for metadata and platform access, view transmission encryption history, view and update application certificates. |

## System settings

System settings involve user permissions, platform settings, APIs, and the full-link dimensional diagnosis feature.

The following table lists the main features of system settings.

| Feature | Description |
| --- | --- |

| Feature | Description |
| --- | --- |
| User permissions | • Users and user groups: allows you to create, modify, and delete custom users and user groups.<br>• Roles: Preset roles such as OAM super administrator, system administrator, security officer, security auditor, and multi-cloud configuration administrator are provided. You can create, modify, and delete custom roles.<br>• Departments: By default, the system has the root department. You can create, modify, and delete departments for the root department, and add users and user groups to departments.<br>• Region authorization: You can bind departments to regions. After you bind a department to a region, users in the department can manage and view resources in the region. You can view authorization information including the authorization version, customer information, authorization type, Elastic Compute Service (ECS) instance ID, cloud platform version, authorization time, and the authorization information of all cloud services in different data centers.<br>• Two-factor authentication: supports two-factor authentication for user logons: account passwords and mobile phone verification codes. You can enable and disable the two-factor authentication feature. Only Google two-factor authentication is supported.<br>• Logon policies: You can view, add, modify, delete, disable, and enable logon policies based on your needs.<br>• Logon settings: You can modify the logon settings to change the logon timeout period, logon policy, and validity period of the current account.<br>• Personal information: You can change the password that you use to log on to the Apsara Uni-manager Operations Console. |
| Platform settings | • Menus: You can hide presets menus. You can add, hide, modify, and delete custom menus.<br>• Authorization: You can view the authorization information of all purchased hybrid cloud software programs, update and view the authorization specifications of software programs at regular intervals, set alert thresholds for authorizing excess or overdue software programs. |
| APIs | • Namespace management: You can view and delete the product information currently registered on the OPSAPI gateway, such as the namespace name and description.<br>• API management: You can view, unpublish, publish, upgrade, and delete APIs for the products registered on the OPSAPI gateway, and upload and register APIs. |

| Feature | Description |
| --- | --- |
| End-to-end | SLA console: This feature is hidden by default.<br>• The product availability sequence diagram, product fault diagnosis, and product dependency information are displayed in the dashboard.<br>• The availability reports of products (both instances and services) and availability reports of the base (both instances and services) are provided. You can view and export availability reports.<br>• End-to-end diagnostics and demarcation is available: forward diagnostic and demarcation information of products, forward diagnostic and demarcation information of resources, reverse influence analysis of products, and reverse influence analysis of services.<br>  ○ Forward diagnostic and demarcation information of products: You can view the diagnostic information of products, including exceptional products, exceptional services, abnormal service roles, dependencies, and root cause analysis. You can view the information about the physical servers where the server roles are deployed and the network topology information of such physical servers. You can diagnose products deployed on the cloud.<br>  ○ Forward diagnostic and demarcation information of resources: You can view the number of resources on the cloud by resource, and view diagnostic information of specified resources such as services, service status, dependencies, and root cause analysis.<br>  ○ Reverse influence analysis of products: You can view dependencies between a product and other products that depend on the product.<br>  ○ Reverse influence analysis of services: You can view dependencies between a service and other services that depend on the service. |
| | End-to-end logs: You can view and export the end-to-end logs of ECS, SLB, and All in ECS. |
| Multi-cloud settings | You can add and modify multi-cloud settings. You can perform O&M on different data centers in one cloud console. You can switch between several cloud consoles.<br>Multi-cloud configuration administrators and super administrators can configure multi-cloud settings. |

# 2.1.4.3. Apsara Uni-manager Dashboards

**Apsara Uni-manager Dashboards provides preset large screens in typical business scenarios, supports flexible customization of large screens, and displays business data in all dimensions.**

**The following table describes the major features of Apsara Uni-manager Dashboards.**

| Feature | Description |
| --- | --- |

| Feature | Description |
| --- | --- |
| Preset dashboards | Five preset dashboards are provided: resources, organizations, security, network, and alerts. They display the running status and resource usage in real time.<br>• Resources dashboard: displays the total quota and usage of resources in each region. The following data is offered:<br>  ○ The total quota and utilization of CPU, and resource usage for ECS.<br>  ○ The inbound and outbound traffic values, and the number of new and active connections for SLB.<br>  ○ The total quota and utilization of CPU, and resource usage for ApsaraDB RDS for MySQL.<br>  ○ The total quota and usage of disks for OSS.<br>  ○ The total quota and VIP usage for VPC.<br>  ○ The total quota and usage of disks for MaxCompute.<br>• Organizations dashboard: displays the total quota and usage of resources for organizations at all levels. The following data is offered:<br>  ○ The total quota and utilization of CPU, and resource usage for ECS.<br>  ○ The inbound and outbound traffic values, and the number of new and active connections for SLB.<br>  ○ The total quota and utilization of CPU, and resource usage for ApsaraDB RDS for MySQL.<br>  ○ The total quota and usage of disks for OSS.<br>  ○ The total quota and VIP usage for VPC.<br>  ○ The total quota and usage of disks for MaxCompute.<br>• Security dashboard: displays the security data at network, application, host, and product levels. The following items are also displayed: the security score for the day, the number of attacks for the week, the number of intercepted DDoS attacks, the number of web application attacks, the seven-day line chart of the intrusion prevention system, and the number of the resources which are not in the secure state.<br>• Network dashboard: displays the network topology, inbound and outbound traffic of network devices, running status of SLB clusters, running status of XGW clusters, port usage, and network device alerts.<br>• Alerts dashboard: displays the total number of alerts for physical servers, network devices, Apsara Stack base, and core products, the number of system alerts, the number of hardware alerts, and the distribution of alerts at different levels. |
| Custom dashboards | The online dashboard editor can be used to create custom dashboards, data, and styles based on templates.<br>• You can set the theme, background, audio, and overall effects of a dashboard.<br>• You can create dashboards based on preset templates, blank templates, and layout templates.<br>• You can set the data source, style, audio, and language of the chart widget, and adjusting the layout of the custom screen.<br>• You can upload and download dashboard page data.<br>• You can preview, modify, view, clone, set, and delete a dashboard. |

| Feature | Description |
| --- | --- |
| Custom dashboard management | • You can create and delete projects, view custom dashboards by project, and rotate custom dashboards.<br>• You can set project names and project descriptions, modify permissions, and view permissions and thumbnails.<br>• You can change dashboard sorting methods in a project. |
| Dashboard-presentation | • You can scan the QR code to switch dashboards to the control terminal.<br>• Dashboards can be displayed in full-screen mode. |

# 2.1.5. Scenarios

This topic describes the scenarios of the Apsara Uni-manager Management Console, Apsara Uni-manager Operations Console, and Apsara Uni-manager Dashboards.

# 2.1.5.1. Apsara Uni-manager Management Console

The scenarios of the Apsara Uni-manager Management Console include hybrid cloud management, multi-level cloud management, heterogeneous cloud management, and industry cloud operations.

## Hybrid cloud management

When customers want to build their own private clouds, the Apsara Uni-manager Management Console can provide tenant-side capabilities such as resource distribution, permission management, and metering and billing management based on enterprise organization models. This enhances resource usage, ensures resource permission security, and improves resource management efficiency. 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.

## 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.

## 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.

## Industry cloud operations

When customers want to build their own cloud platforms, the Apsara Uni-manager Management Console can deliver service sales and operations by providing personalized configurations and process provisioning.

# 2.1.5.2. Apsara Uni-manager Operations Console

This topic describes the scenarios of the Apsara Uni-manager Operations Console: routine O&M, automated O&M, security O&M, and remote O&M.

## Routine O&M

**Scenarios:** If you want to view alerts and inventory in real time during daily maintenance, you can view information on the alert dashboard and Inventory Analysis page.

- Alert monitoring: View alerts on the alert dashboard. In multi-region scenarios, you can quickly view alerts in each region. On the Alerts page, you can view alert details to discover alert sources. Solution documents are provided for some common alerts to help quickly handle alerts.
- Inventory management: On the Inventory Analysis page, you can view the inventory and forecast usage of resources for inventory analysis and decision-making.

**Benefits:**

- Centralized monitoring of alerts allows you to check the health of resources in real time, quickly locate and handle alerts, and therefore improve O&M efficiency.
- Centralized monitoring of resource inventory and forecasts of usage trends allow you to adjust resource distribution.

## Automated O&M

**Scenarios:** If you want to implement automated O&M by orchestrating complex O&M actions or typical O&M scenarios such as scheduled inspection of physical machines, scheduled data backup, batch execution of commands, and batch update of software packages, you can use O&M scripts and custom O&M jobs and orchestrate O&M processes on the Automated O&M page to perform scheduled or immediate O&M tasks.

**Benefits:** A wide range of O&M script libraries and flexible O&M orchestration processes allow you to standardize typical O&M scenarios and simplify complex O&M scenarios, save O&M and labor costs, avoid boring and repetitive tasks, and enhance O&M efficiency and satisfaction.

## Security O&M

**Scenarios:** If you want to control the security of virtual machines or databases where server roles are deployed, you can log on to virtual machines or databases with one click on the Security O&M page to audit CLI operations, view risk alerts, and block high-risk operations.

**Benefits:**

- You can log on to VMs and databases with one click and download O&M logs. This simplifies logon procedures and improves O&M efficiency.
- Risk alerts and blocking high-risk operations can reduce system or service failures caused by accidental operations. Audit records conduce to discovering root causes and improving system security and stability.

### Remote O&M

**Scenarios:** If you need an expert team from Apsara Stack Online to provide 24/7 O&M, you can configure the IP address and port number of the worker and the IP address of Apsara Stack Online on the Security page for connecting to Apsara Stack Online.

**Benefits:**

- You are relieved from complex O&M of underlying platforms and better focus on cloud business.
- The professional O&M team is on duty in a 24/7 manner and provides O&M services such as centralized monitoring, risk governance, intelligent inspection, unified authentication, upgrades & changes, and operation auditing. This can discover and solve problems in a timely manner, improve O&M efficiency, and reduce O&M costs.

## 2.1.5.3. Apsara Uni-manager Dashboards

Apsara Uni-manager Dashboards monitors the overall running status and health of resources. For presentations in exhibition halls or operation centers, you can quickly switch dashboards to the control terminal. Business and O&M visualization assists decision-making and improves online business efficiency and O&M capabilities.

### Global monitoring

In routine O&M, you can use present dashboards and custom dashboards to monitor the running status and usage of resources in real time.

### Dashboard presentation

To display the running status of resources in exhibition halls or operation centers, you can switch real-time dashboards to the dashboard presentation mode for centralized monitoring, presentation, or reporting.

# 2.1.6. Limits

N/A.

# 2.1.7. Terms

### Public cloud

A deployment model in which the infrastructure is owned by an organization and provides cloud computing services to the public or an industry.

### Private cloud

A cloud computing service deployment mode that deploys cloud infrastructure and software and hardware resources in an internal network for organizations or departments within an enterprise.

### Hybrid cloud

A cloud computing service deployment model that combines public and private clouds.

# 3.Elastic Compute Service (ECS)

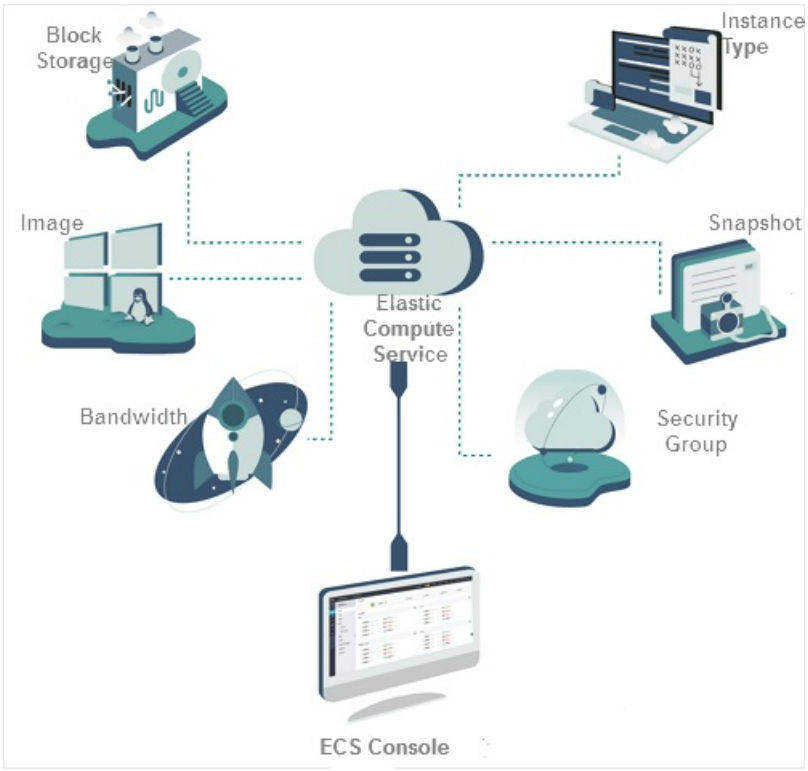
## 3.1. Product Introduction

### 3.1.1. 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




### 3.1.2. Benefits

Elastic Compute Service (ECS) is easier to use and provides higher availability, security, and elasticity than services provided by other server vendors and traditional Internet data centers (IDCs).

#### High availability

Compared with traditional servers that are limited by hardware, ECS provides higher O&M standards and integrates the features of a variety of cloud services to offer more efficient backup, disaster recovery, and failover.

Apsara Stack provides support in the following areas:

- Industry and ecosystem partners to help you build a more advanced and stable architecture.
- 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 cloud service users. Alibaba Cloud has received a host of international information security certifications that impose strict requirements for the 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 within a VPC.
- **Comprehensive security protection** is provided for ECS and includes security policies, security hardening, data security, and monitoring and alerts to improve the security of your business and defend against external attacks and unauthorized access.

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 change 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 sufficient resources during peak hours. When the load returns to normal levels, you can release redundant resources to reduce 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 upgrade server configurations or replace servers to increase the storage space. In Apsara Stack, you can resize attached disks or attach more disks to instances to increase the storage space.

- **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, network as well as business architecture planning. You can use Apsara Stack to build your business portfolio based on your needs.

### Ease of use

Apsara Stack provides an easy-to-use console for centralized management. You can use the console to perform operations on ECS instances and ECS-related services to have ECS instances created, related services activated, and the configurations of the instances and services modified. Aspara Stack provides the following resources to help you use it in an easy way:

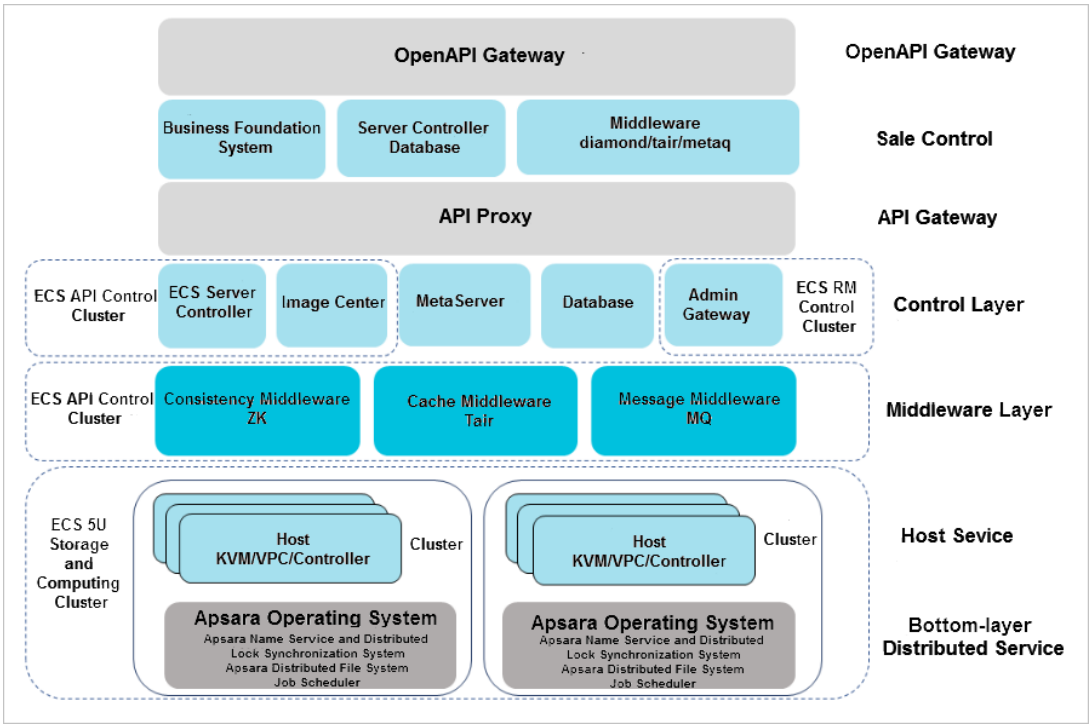
- Diverse instance families such as shared instance families, dedicated instance families, and ECS Bare Metal Instance families.
- VPCs that serve various purposes. You can choose VPCs based on your business requirements and your organization.
- A variety of storage types. You can choose storage types based on your requirements.
- Ease-to-use security policies.

Apsara Stack also allows you to use images, deploy clusters, use custom tags, migrate virtual machines, and change instance configurations.

## 3.1.3. Architecture

Elastic Compute Service (ECS) is built on the Apsara distributed operating system developed by Alibaba Cloud. Individual ECS instances are virtualized by using Kernel-based Virtual Machine (KVM) or the SHENLONG architecture developed by Alibaba Cloud. ECS uses Apsara Distributed File System for data storage.

### 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. By 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. |
| **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, creates snapshot policies, processes sales requests, and provides APIs to users. |
| **API Proxy** | Forwards requests to services that are deployed in the region specified by regionid. |

| Component | Description |
| --- | --- |
| 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. |
| Message Queue | Provides message queues for the status 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, Virtual Private Cloud (VPC) for network virtualization, and control through interaction with Libivrt. |
| 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. |

# 3.1.4. Features

## 3.1.4.1. Instances

### 3.1.4.1.1. Overview

An Elastic Compute Service (ECS) instance is the smallest computing service unit in the cloud. An instance type essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute, network, and storage capabilities. An ECS instance is a virtual server that includes basic components such as CPUs, memory, an operating system (OS), network configurations, and disks. You can use management tools provided by Alibaba Cloud such as the ECS console and ECS API to create and manage ECS instances. You can manage the status of ECS instances and their deployed applications in the same manner as you would do with local servers. You can also upgrade the compute, network, and storage capabilities of your ECS instances as needed to meet increasing requirements.

### 3.1.4.1.2. Instance families

Alibaba Cloud Elastic Compute Service (ECS) instances are categorized into different instance families based on the business and application scenarios for which they are suited. ECS instance families include shared instance families, dedicated instance families, ECS Bare Metal Instance families, instance families equipped with local HDDs, instance families equipped with local SSDs, heterogeneous computing instance families, and Super Computing Cluster (SCC) instance families. Each instance family consists of one or more instance types that share similar attributes.

## Instance families

The following table describes the instance families and their application scenarios.

| Instance family | Application scenario |
| --- | --- |
| Shared instance family | Shared instance families are suitable for business scenarios that do not require high performance of virtual machines, such as small and medium-sized websites and web applications, development environments, servers, code repositories, microservices, testing and staging environments, lightweight databases, lightweight enterprise applications, and integrated application services. |
| Dedicated instance family | Dedicated instance families are suitable for business scenarios that require high performance of virtual machines, such as web frontend servers, data analysis, batch processing, video encoding, high-performance scientific and engineering applications, and scenarios where large volumes of packets are received and transmitted. |
| ECS Bare Metal Instance family | ECS Bare Metal Instance families provide robust compute, storage, and network configurations for business scenarios that have high requirements for dedicated resources, security isolation, and performance, such as containers, databases, core business of enterprises, and big data computing. |
| Instance family equipped with local HDDs | Instance families equipped with local HDDs are suitable for business scenarios that have high requirements for big data computing, storage, and analysis, such as mass storage and offline computing scenarios, and can meet the high requirements of distributed computing services such as Hadoop in terms of storage performance, storage capacity, and internal bandwidth. |
| Instance family equipped with local SSDs | Instance families equipped with local SSDs are suitable for I/O-intensive applications that require high I/O performance and low latency, such as NoSQL databases (including Cassandra and MongoDB), MPP data warehouses, distributed file systems, and search scenarios that use solutions such as Elasticsearch. |
| Heterogeneous computing instance family | Heterogeneous computing instance families use various accelerators including NVIDIA T4 GPUs, NVIDIA V100 GPUs, and FPGAs, and are suitable for business scenarios such as AI inference, computer vision, speech recognition, speech synthesis, machine translation, recommendation systems, real-time rendering, deep learning, and scientific computing applications. |
| SCC instance family | SCC instance families provide computing cluster services with ultimate computing performance and parallel efficiency by using the high-speed InfiniBand (IB) network on top of ECS Bare Metal Instance families. SCC instance families are suitable for scenarios such as high-performance computing, artificial intelligence (AI), machine learning, scientific and engineering computing, data analysis, and audio and video processing. |

## Resource allocation

ECS instances of different types vary in resource allocation.

- For all ECS instances that use Kernel-based Virtual Machine (KVM), CPU and memory resources must be reserved for virtualization. For example, you can reserve 8 vCPUs and 32 GiB of memory, 10 vCPUs and 40 GiB of memory, or 10 vCPUs and 48 GiB of memory to handle network virtualization, storage virtualization, and system control tasks.
- For all ECS instances that use the SHENLONG architecture, no CPU or memory resources need to be reserved for the server. The SHENLONG architecture integrates software and hardware and offloads network virtualization, storage virtualization, and system control to a dedicated chip system to achieve zero resource overheads and prevents the performance of virtual machines from fluctuating when demand outstrips resource supplies at the underlying layer.
- Resources of the memory and local disks in hosts cannot be overprovisioned.
- CPU resources of hosts that host shared and burstable instances can be overprovisioned. We recommend that you keep the overprovisioning ratio below 4. CPU resources of hosts that host other instances cannot be overprovisioned.

> **Note** For more information about the supported instance families, see *Instance families and instance types*.

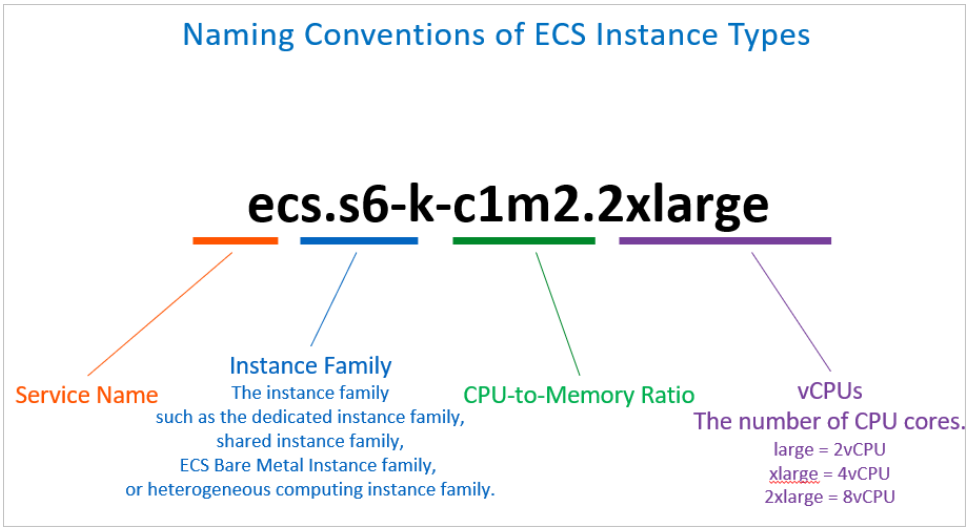
## 3.1.4.1.3. Instance types

An Elastic Compute Service (ECS) instance family can be categorized into multiple instance types based on their vCPU and memory configurations.

ECS instance type defines the specific attributes of instances, such as the number of vCPUs, memory size, network capability (including network bandwidth of the virtual private cloud (VPC), packet forwarding rate, the number of elastic network interfaces (ENIs), and the number of IP addresses per ENI), and storage capability (maximum disk bandwidth, maximum disk IOPS, and the number of attached disks). The network and storage performance of instances in the same instance family depends on their computing capacities. The instances provide high network and storage performance based on large computing capacity.

The following section Naming conventions of ECS instance types describes the naming conventions of ECS instance types.

Naming conventions of ECS instance types




The following section describes the naming conventions of ECS instance types.

- Service Name: The name starts with ecs.,which indicates ECS.
- Instance Family: For example, in s6-k, s indicates the shared instance family, 6 indicates the sixth-generation instance family, and k indicates Kernel-based Virtual Machine (KVM) virtualization. In g6x-k10, g indicates the dedicated instance family, 6 indicates the sixth-generation instance family, x indicates the use of 128 logic cores, and k10 indicates the use of KVM and 10 GB network.
- CPU-to-Memory Ratio: For example, c1m2 indicates a CPU-to-memory ratio of 1:2, such as a configuration of 2 vCPUs and 4 GiB memory, or 4 vCPUs and 8 GiB memory.
- vCPUs: For example, large indicates 2 vCPUs, xlarge indicates 4 vCPUs, 2xlarge indicates 8 vCPUs, and 3xlarge indicates 12 vCPUs.

> **Note** For more information about the supported instance types, see *Instance families and instance types*.

# 3.1.4.1.4. 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

## 3.1.4.1.5. Instance lifecycle

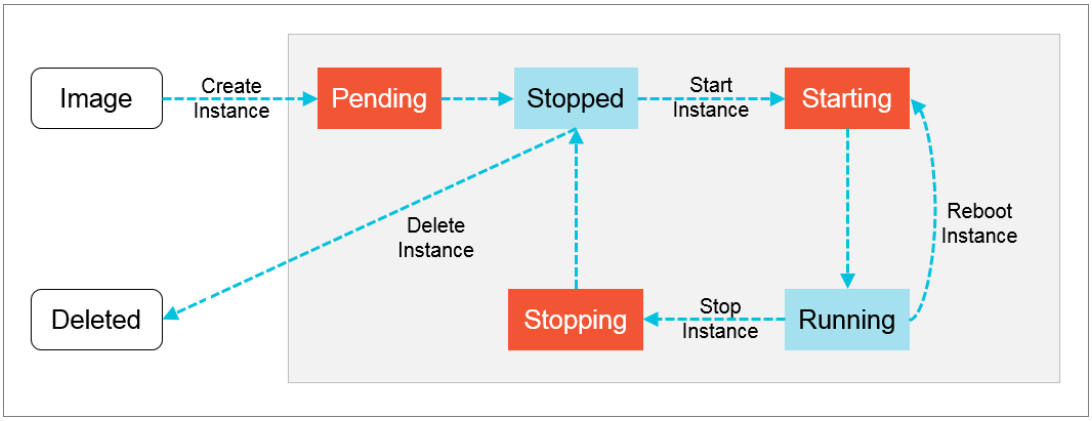
The lifecycle of an ECS instance begins when the instance is created and ends when the instance is released. This topic describes the instance states in the ECS console as well as state attributes and their corresponding instance states in API responses.

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 |

| State | State attribute | Description | State in an API response |
| --- | --- | --- | --- |
| 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.




# 3.1.4.1.6. ECS Bare Metal Instance families:

Elastic Compute Service (ECS) Bare Metal Instance is an innovative cloud computing service developed by Alibaba Cloud on top of SHENLONG architecture that integrates hardware and software. ECS Bare Metal Instance combines virtual machine features (such as elasticity of resources, resource delivery within minutes, and automated O&M) and physical machine features (such as performance consistency, hardware feature sets, and strong hardware-level isolation). ECS Bare Metal Instance is fully compatible with services in the Alibaba Cloud ecosystem and can help enterprises migrate their critical and high-load applications to the cloud.

ECS Bare Metal Instance utilizes technical innovation to meet your business requirements. It has the following features:

- **Security, reliability, and high performance**

ECS Bare Metal Instance combines the elasticity of virtual machines with physical machine features such as performance consistency, hardware feature sets, high security, and strong hardware-level isolation. ECS Bare Metal Instance reserves ECS resources for the exclusive use by a single tenant and meets your business requirements for instance performance and stability as well as data protection and compliance regulation.

- **Compatibility with the ECS technology system**

  ECS Bare Metal Instance is compatible with the virtual private clouds (VPCs), disks, images, and management systems of ECS. You can call API operations, use the ECS console, and use Virtual Network Computing (VNC) to create and manage ECS bare metal instances as you would do with other ECS instances. In Apsara Stack, instances can be migrated between ECS Bare Metal Instance and other ECS instances. No communication bottlenecks exist between ECS Bare Metal Instance clusters and other ECS instance clusters within VPCs.

- **Delivery within minutes**

  The auto scaling feature of ECS is the core feature of cloud services at the IaaS layer. ECS bare metal instances share resource pools with and have the same auto scaling capabilities as other ECS instances. ECS bare metal instances can be delivered within minutes in respond to unexpected business needs.

- **Automated O&M**

  Similar to other ECS instances, ECS bare metal instances provide automated O&M capabilities. If the host of an ECS bare metal instance that does not use local disks fail, the O&M management system migrates the instance to a normal host to minimize service interruptions.

- **Compatibility with other services**

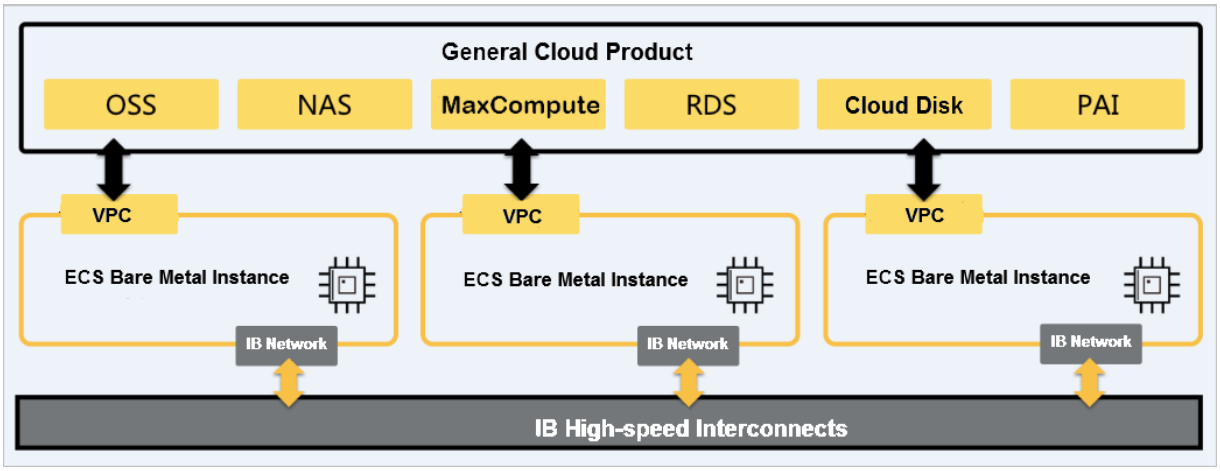
  ECS Bare Metal Instance is compatible with services in the Alibaba Cloud ecosystem, such as Container Service for Kubernetes (ACK), Server Load Balancer (SLB), and ApsaraDB RDS and provides a complete set of solutions to meet your business needs in different workload scenarios.

# 3.1.4.1.7. 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. |

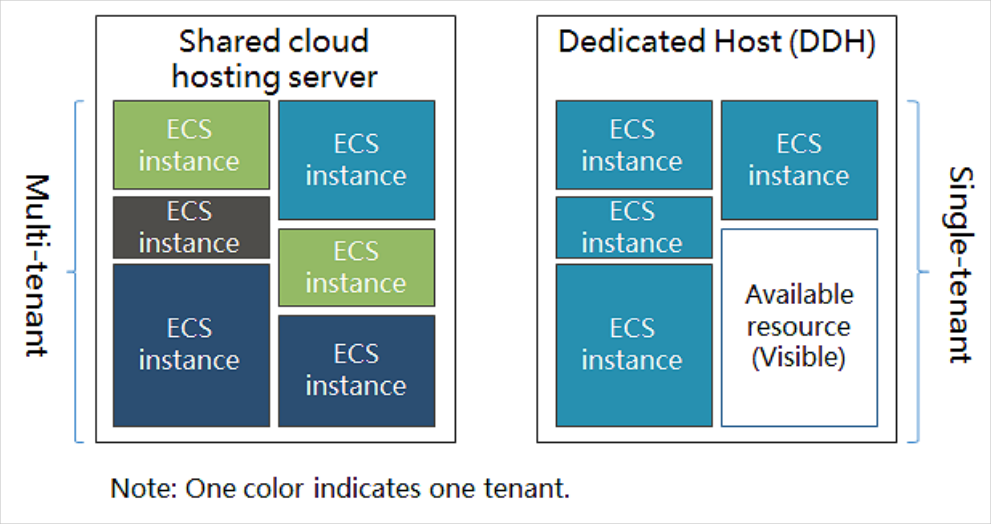
| Scenario | Description |
| --- | --- |
| Supercomputing center in the cloud | Supercomputing centers are the earliest and most mature IT service developed in China. Traditional supercomputing centers integrate technologies from multiple vendors, which requires lengthy and complex technical solution consultation, equipment selection, and verification. The construction cycle is long, and operations and maintenance are complex.<br>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. |
| 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:<br>• Petrochemical: seismic data processing and reservoir simulation<br>• Finance: financial derivative analysis, actuarial analysis, asset and liability management, and investment risk analysis<br>• Industrial manufacturing: collision analysis, failure analysis, and thermodynamic analysis<br>• Life science: drug discovery, protein folding, DNA sequencing, and medical imaging<br>• Media and entertainment: video post-production and animation rendering<br>• Government and higher education: weather forecast, high-energy physics, and geophysics |

# 3.1.4.1.8. Dedicated hosts

Dedicated Host is a cloud solution customized and optimized by Apsara Stack for enterprise users. Dedicated hosts offer high cost effectiveness and physical resources for exclusive use and can be used to deploy and configure your business in a more flexible and comprehensive manner.

A dedicated host is a cloud host whose physical resources are reserved for the exclusive use by a single tenant. As the only tenant of a dedicated host, you do not need to share its physical resources with other tenants. You can obtain the physical attributes of the host, including the number of sockets (CPUs), the number of physical CPU cores, and memory size. You can also create Elastic Compute Service (ECS) instances of a specified instance family that is compatible with the host type.

Unlike shared hosts whose resources are shared by multiple tenants, dedicated hosts provide a dedicated hosting environment for a single tenant based on the virtualization technology of Apsara Stack. Dedicated hosts offer flexible and scalable services that enable you to exclusively use all the physical resources of a cloud host. The following figure shows the differences between dedicated hosts and shared hosts.




ECS instances that run on dedicated hosts have the same performance as those that run on shared hosts.

# 3.1.4.2. Block storage

# 3.1.4.2.1. 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 |
| --- | --- | --- |
| 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. |

# 3.1.4.2.2. Elastic block storage

# 3.1.4.2.2.1. Overview

Elastic block storage can be divide 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.

# 3.1.4.2.2.2. Disks

Disks are block-level storage devices provided by Apsara Stack for Elastic Compute Service (ECS) instances. Disks can be classified based on their performance or purposes.

## Performance-based classification

Disks can be classified into ultra disks, shared ultra disks, standard SSDs, shared standard SSDs, standard performance disks, and premium performance disks based on their performance.

- Ultra disks and shared ultra disks are ideal for medium I/O load scenarios and deliver up to 5,000 random IOPS.
- Standard SSDs and shared standard SSDs are ideal for I/O-intensive scenarios and deliver up to 25,000 random IOPS.
- Standard performance disks and premium performance disks are ideal for online transaction processing (OLTP) databases and NoSQL databases and deliver up to 25,000 random IOPS.

> **Important** Different Elastic Block Storage (EBS) clusters support different disk categories.
>
> - Newly deployed EBS clusters in Cloud Defined Storage (CDS) support premium performance disks and standard performance disks.
> - EBS clusters in CDS that are created in Apsara Stack V3.15.0 and earlier support ultra disks, standard SSDs, premium performance disks, and standard performance disks.
> - Existing EBS clusters continue to provide shared ultra disks and shared standard SSDs.

The following table compares the performance of different disk categories.

| Category | Standard SSD and shared standard SSD | Ultra disk and shared ultra disk | Standard performance disk | Premium performance disk |
|---|---|---|---|---|
| Maximum capacity of a single disk (GiB) | 32768 GiB | 32768 GiB | 32768 GiB | 32768 GiB |
| Maximum IOPS | 25000 | 5000 | 5000 | 25000 |
| Maximum throughput (MB/s) | 300 MB/s | 140 MB/s | 140 MB/s | 300 MB/s |
| Formula for calculating the IOPS per disk | min{1800 + 30 × Capacity, 25000} | min{1800 + 8 × Capacity, 5000} | min{1800 + 8 × Capacity, 5000} | min{1800 + 30 × Capacity, 25000} |
| Formula for calculating the throughput per disk (MB/s) | min{120 + 0.5 × Capacity, 300} | min{100 + 0.15 × Capacity, 140} | min{100 + 0.15 × Capacity, 140} | min{120 + 0.5 × Capacity, 300} |
| API parameter value | cloud_ssd | cloud_efficiency | cloud_sperf | cloud_pperf |

| Category | Standard SSD and shared standard SSD | Ultra disk and shared ultra disk | Standard performance disk | Premium performance disk |
| --- | --- | --- | --- | --- |
| Use scenario | Small and medium-sized development and test applications that require high data durability | • Development and testing applications<br>• System disks | • OLTP databases: relational databases such as MySQL, PostgreSQL, Oracle, and SQL Server databases<br>• NoSQL databases: non-relational databases such as MongoDB, HBase, and Cassandra databases<br>• Elasticsearch distributed logs: Elasticsearch, Logstash, and Kibana (ELK) log analysis | |

### Purpose-based classification

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

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

## 3.1.4.2.2.3. Shared disks

Shared disks are block-level data storage devices that feature high concurrency, high performance, and high reliability. Shared disks support concurrent reads and writes to multiple Elastic Compute Service (ECS) instances.

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

Shared disks can be classified into the following types based on their performance:

- **Shared standard SSD**: uses SSDs as the storage medium to provide stable and high performance storage that offers enhanced random I/O and data durability.
- **Shared ultra disk**: uses a hybrid SSD and HDD storage medium.

Cloud disks and shared disks can be used as data disks and up to 16 data disks can be attached to a single instance.

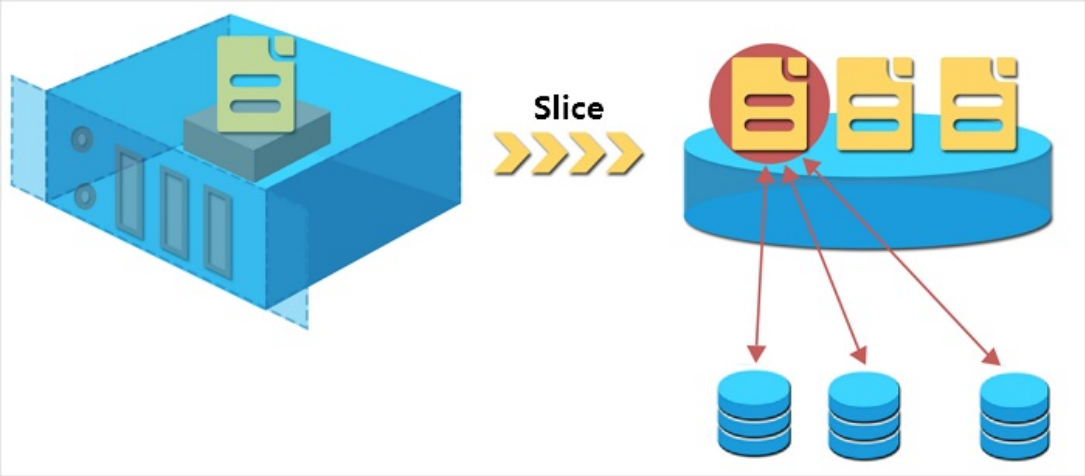
## 3.1.4.2.2.4. Triplicate storage

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:

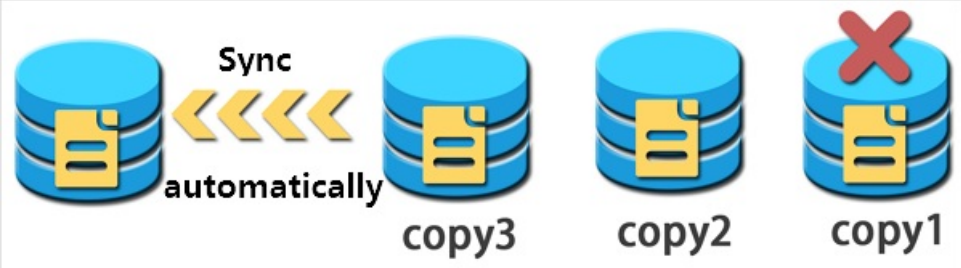
1. The client determines the location of a chunk corresponding to the write operation.
2. The client sends a request to the master to query the chunk servers where the three chunk replicas are each stored.
3. The client sends write requests to the chunk servers based on the results returned from the 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 ensure that chunk replicas are each distributed to chunk servers on different racks and that data does not become unavailable due to the failure of a single server or rack.

## Data protection mechanism

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

Automatic replication




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

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

# 3.1.4.2.2.5. 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)` :<br>When m is 8 and n is 3, the storage usage is calculated based on the following formula: 8/(8 + 3) = 72.7%. | 1/3 = 33.3% |
| Reliability | Allows up to n fragments to be lost. Failures on up to n servers are allowed in the worst case. For example, when m is 8 and n is 3, failures on up to three servers are allowed. | Allows up to two replicas to be lost. Failures on up to two servers are allowed in the worst case. |

# 3.1.4.2.3. ECS disk encryption

Elastic Compute Service (ECS) disk encryption is a simple and secure encryption method that can be used to encrypt new 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 negative impacts on your business. The following types of data can be encrypted:

- Data stored on disks.
- Data transmitted between disks and instances. Data within the instance operating system is not encrypted.
- All snapshots created from encrypted disks. These snapshots are encrypted snapshots.

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

Ultra disks, shared ultra disks, standard SSDs, shared standard SSDs, premium performance disks, and standard performance disks can be encrypted.

# 3.1.4.2.4. ECS disk resizing

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

## Scenarios

You can use one of the following methods to increase the storage capacity of a single instance:

- Resize an existing disk. You can resize the existing partitions of the disk or create more partitions on the disk.

  The following table describes the two methods for resizing an existing disk.

| Method | Limits |
| --- | --- |
| Resize a disk online | The instance to which the disk is attached must be in the **Running** (*Running*) state.<br>After you resize the disk, the new size takes effect without the need to restart the instance.<br>For more information about limits on disk resizing and how to resize a disk online, see *Resize disks* in the *ECS User Guide*. |

| Method | Limits |
| --- | --- |
| Resize a disk offline | The instance to which the disk is attached must be in the **Running** (*Running*) or **Stopped** (*Stopped*) state.<br>After you resize the disk, you must restart the instance by using the Elastic Compute Service (ECS) console or by calling the RebootInstance operation for the new size to take effect.<br>For more information about limits on disk resizing and how to resize a disk offline, see *Resize disks* in the *ECS User Guide*. |

- Create a disk, attach the disk to an ECS instance as a data disk, and then partition and format the disk.
- Replace the system disk of an instance and specify a larger size for the new system disk.

### Size ranges of resized system disks

The new size of a resized system disk must be larger than the original size but cannot exceed 2,048 GiB.

For example, the system disk of a CentOS instance is 35 GiB in size. When you resize this system disk, the specified new size must be larger than 35 GiB but cannot exceed 2,048 GiB.

### Maximum size of a resized data disk

The new size of a resized data disk must be larger than the original size. Ultra disks, shared ultra disks, standard SSDs, shared standard SSDs, premium performance disks, and standard performance disks can be resized up to 32,768 GiB.

## 3.1.4.2.5. 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.

# 3.1.4.3. Images

An image is a template for running environments within Elastic Compute Service (ECS) instances. An image includes an operating system and pre-installed applications.

An image works as a copy that stores data from one or more disks. An 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 the following Windows Server operating systems and mainstream Linux operating systems. Different platforms support different images.<br>• Intel x86<br>  ○ Windows<br>  ○ CentOS<br>  ○ Debian<br>  ○ FreeBSD<br>  ○ OpenSUSE<br>  ○ SUSE Linux<br>  ○ Ubuntu<br>  ○ Anolis OS<br>  ○ Alibaba Cloud Linux 2<br>• Hygon x86<br>  ○ UOS<br>  ○ Kylin<br>  ○ NFS<br>• ARM<br>  ○ UOS<br>  ○ Kylin<br>  ○ CentOS<br>  ○ Anolis OS<br>  ○ Aliyun Linux 2 |

| Type | Description |
| --- | --- |
| Custom Image | Custom images are created from ECS instances or snapshots or imported from your computer. Custom images can contain applications and data. You can use custom images to create instances that have identical configurations. This eliminates the need to make repeated configurations. |

### Obtain an image

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

- Create a custom image based on an existing ECS instance.
- Use 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 deploy identical 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 about format conversion, see "Convert the image file format" in *ECS User Guide*.

# 3.1.4.4. Snapshots

# 3.1.4.4.1. 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:

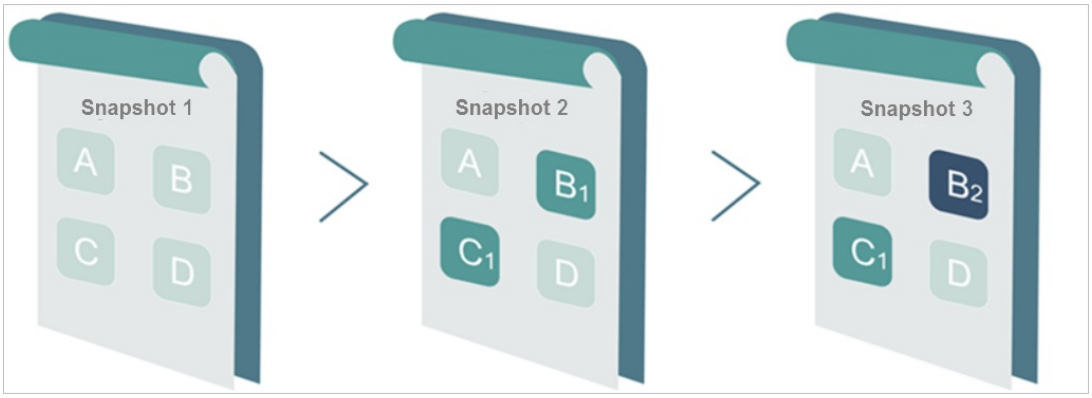
- You may want to use the data of one disk as the basis to write or store data to a different disk. To achieve this, you can create a snapshot for a cloud disk and then create another cloud disk from the snapshot. The new disk contains the basic data of the original disk.
- While cloud disks are a secure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

# 3.1.4.4.2. 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.

# 3.1.4.4.3. 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 |
|---|---|---|---|---|

| 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.<br>• A snapshot is created for the data disks of core business every four hours. Snapshots taken within the last ten days are retained. |
| Automatic snapshot policy | By default, the task is scheduled to be triggered once a day and cannot be modified manually. | You can customize the time of day and days of the week that snapshots are scheduled to be created and the retention period of snapshots. The disk quantity and related details associated with an automatic snapshot policy can be queried. | More flexible protection policy. | • You can schedule snapshots to be created on the hour several times in a single day.<br>• You can specify the days of the week for which to create snapshots.<br>• You can specify the snapshot retention period or choose to retain a snapshot permanently. When the number of automatic snapshots reaches the upper limit, the oldest automatic snapshot will be automatically deleted. |
| Implementation | Copy-on-write (COW) | Redirect-on-write (ROW) | Mitigates the impact of snapshot tasks on business I/O performance. | Snapshots can be taken at any time without interruptions to your business. |

# 3.1.4.5. 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
  - StrictlAggregation
  - 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.

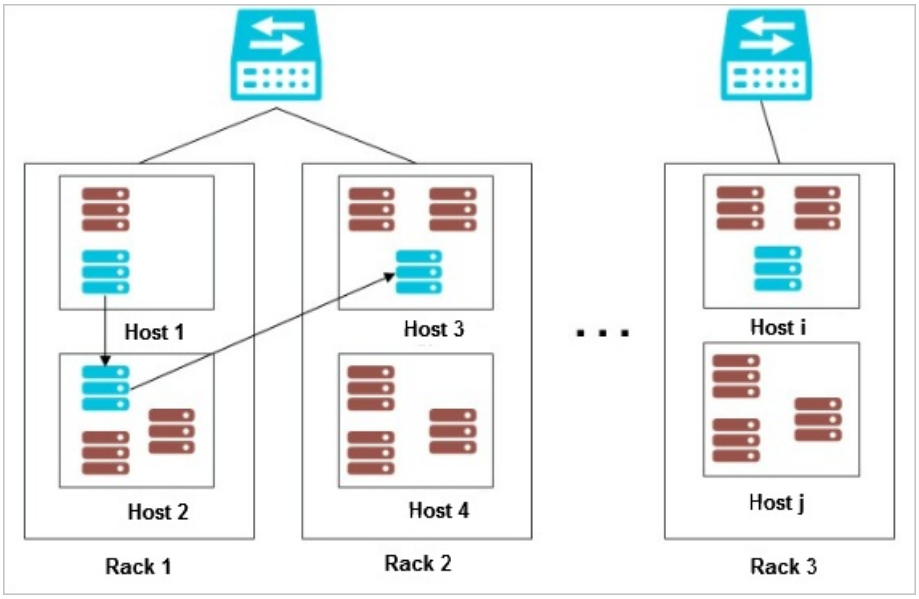
Granularities and policies

| 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* .

# 3.1.4.6. Network and security

# 3.1.4.6.1. 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.

# 3.1.4.6.2. ENIs

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

## Introduction

An ENI is a virtual network interface controller (NIC) that can be bound to an Elastic Compute Service (ECS) instance in a virtual private cloud (VPC). You can use ENIs to deploy high-availability clusters and perform low-cost failover and fine-grained network management.

## Use scenarios

ENIs are suitable for the following scenarios:

- Deployment of high-availability clusters

  Multiple ENIs can be bound to a single ECS 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 redirect traffic destined for the failed instance to the normal instance and immediately recover 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 to isolate management data from business data. You can also configure security group rules for each ENI based on the source IP address, protocols, and ports to implement access control.

## ENI types

ENIs are classified into the following 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 the primary ENI cannot be unbound 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 secondary ENIs that can be bound to a single instance varies based on the instance type.

## ENI attributes

The following table describes the attributes of an ENI.

Attribute description

| Attribute | Quantity |
| --- | --- |
| Primary private IP address | 1 |
| MAC address | 1 |
| Security group | 1 to 5 |
| ENI Description | 1 |
| ENI Name | 1 |

### Limits

The following limits apply to ENIs:

- The number of ENIs that can be created for one account is limited in each region. The ENI quota is displayed on the **Quota overview** page of the Apsara Uni-manager Management Console.
- The ENI and the instance to which the ENI is bound must reside within the same zone of the same VPC, but can be connected to different vSwitches.
- Only specific instance types support ENIs. The maximum number of ENIs that can be bound to a single instance varies based on the instance type.
- The bandwidth of an instance cannot be increased by binding multiple ENIs to the instance.

> **Note** The bandwidth of an instance varies based on the instance type.

## 3.1.4.6.3. 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.

# 3.1.4.6.4. 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.

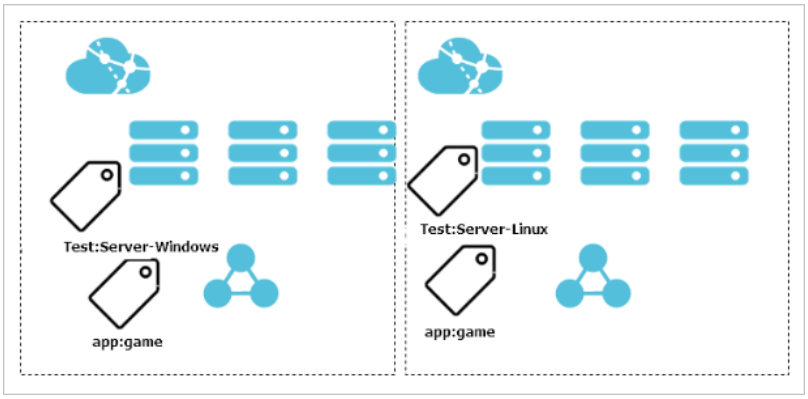
# 3.1.4.7. Tags

Tags allow enterprises and individuals to identify and categorize their Elastic Compute Service (ECS) resources and simplify the query and management of the resources.

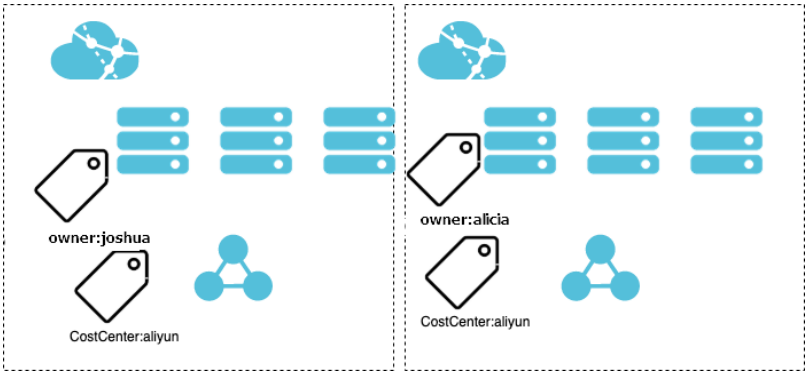
## Use scenarios

When you have multiple ECS instances, you can add tags to ECS resources to categorize and manage the resources and use the tags to search for and batch manage the resources. For example, you can use tags to batch change applications that are deployed by using images, update patches, and configure security group rules that control network access. Tags can be used in the following scenarios:

- You can add different tags to environments such as production and test environments, operating systems such as Windows and Linux, and mobile platforms such as iOS and Android. For example, you can add the `Test:Server-Windows` tag to all Windows ECS instances in the test environment. During daily O&M, you can find these instances based on the tag and perform batch operations on these instances.




- In team or project management, you can add tags such as `CostCenter:aliyun` to groups, projects, or departments. Then, you can categorize the groups, projects, or departments and implement cross authorization based on the tags.




### Usage notes

- Each tag is a key-value pair.
- A tag must have a unique tag key.

  For example, the `city:shanghai` tag is added to an instance. If a new `city:hangzhou` tag is added to the instance, the `city:shanghai` tag is automatically removed from the instance.
- Tags are not shared across regions.
- If a tag is not added to a resource after the tag is removed from a resource, the tag is automatically deleted.

## 3.1.5. 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.

# 3.1.6. 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.

# 3.2. Introduction to Instance Families and Instance Types

## 3.2.1. Instance families and instance types

Alibaba Cloud Elastic Compute Service (ECS) instances are categorized into different instance families based on the business and use scenarios for which they are suited. ECS instance families include shared instance families, dedicated instance families, ECS Bare Metal Instance families, instance families equipped with local HDDs, instance families equipped with local SSDs, heterogeneous computing instance families, and Super Computing Cluster (SCC) instance families. Each instance family consists of one or more instance types that share similar attributes.

### Instance families

The following table describes the instance families and their use scenarios.

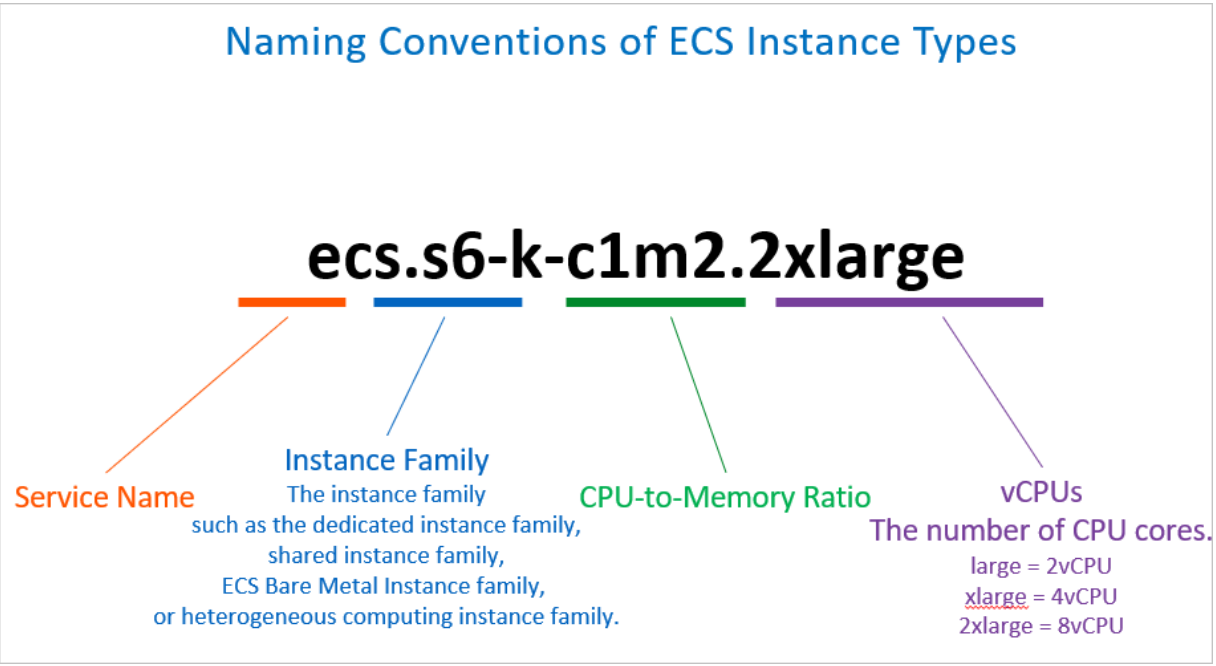
| Instance family | Use scenario |
| --- | --- |
| Shared instance families | Shared instance families are suitable for business scenarios where performance requirements are not high, such as small and medium-sized websites and web applications, development environments, servers, code repositories, microservices, testing and staging environments, lightweight databases, lightweight enterprise applications, and integrated application services. |

| Instance family | Use scenario |
| --- | --- |
| Dedicated instance families | Dedicated instance families are suitable for business scenarios that require high performance, such as web frontend servers, data analysis, batch processing, video encoding, high-performance scientific and engineering applications, and scenarios where large volumes of packets are received and transmitted. |
| ECS Bare Metal Instance families | ECS Bare Metal Instance families provide robust compute, storage, and network configurations for business scenarios that require dedicated resources, security isolation, and high performance, such as containers, databases, core business of enterprises, and big data computing. |
| Instance families equipped with local HDDs | Instance families equipped with local HDDs are suitable for business scenarios such as mass storage and offline computing that have high requirements for big data computing, storage, and analysis and can meet the high requirements of distributed computing services such as Hadoop in terms of storage performance, storage capacity, and internal bandwidth. |
| Instance families equipped with local SSDs | Instance families equipped with local SSDs are suitable for I/O-intensive applications that require high performance and low latency, such as NoSQL databases (including Cassandra and MongoDB), MPP data warehouses, distributed file systems, and search scenarios that use solutions such as Elasticsearch. |
| Heterogeneous computing instance families | Heterogeneous computing instance families use various accelerators including NVIDIA T4 GPUs, NVIDIA V100 GPUs, and FPGAs, and are suitable for business scenarios such as artificial intelligence (AI) inference, computer vision, speech recognition, speech synthesis, machine translation, recommendation systems, real-time rendering, deep learning, and scientific computing applications. |
| SCC instance families | SCC instance families provide computing cluster services with ultimate computing performance and parallel efficiency by using the high-speed InfiniBand network on top of ECS Bare Metal Instance families. SCC instance families are suitable for scenarios such as high-performance computing, AI, machine learning, scientific and engineering computing, data analysis, and audio and video processing. |
| Burstable instance families | Burstable instance families provide a baseline level of CPU performance with the ability to burst above the baseline by spending CPU credits. They are suitable for business scenarios where performance requirements are not high, such as web applications and small and medium-sized websites. |

## Instance types

An instance family can be categorized into multiple instance types based on their vCPU and memory specifications. ECS instance types define the basic attributes of instances, such as the number of vCPUs, memory size, network capabilities, and storage capabilities. Network capabilities include the network bandwidth, packet forwarding rate, maximum number of elastic network interfaces (ENIs) per instance, and maximum number of IP addresses per ENI. Storage capabilities include the maximum disk bandwidth, maximum disk IOPS, and maximum number of attached disks per instance. The network and storage performance of instances within the same instance family depend on their computing capacities. The larger computing capacity the instances have, the higher network and storage performance the instances can deliver.

The following figure shows the naming conventions of ECS instance types.




Each instance type name consists of the following parts:

- Service name: Each instance type name starts with ecs., which indicates ECS.
- Instance family name: Examples: s6-k and g6x-k10. In s6-k, s indicates the shared instance family, 6 indicates the sixth generation, and k indicates Kernel-based Virtual Machine (KVM) virtualization. In g6x-k10, g indicates the dedicated instance family, 6 indicates the sixth generation, x indicates the use of 128 logical cores, and k10 indicates the use of KVM and 10 Gbit/s networks.
- CPU-to-memory ratio: For example, c1m2 indicates a CPU-to-memory ratio of 1:2, such as a configuration of 2 vCPUs and 4 GiB of memory, or 4 vCPUs and 8 GiB of memory.
- Number of vCPUs: For example, large indicates 2 vCPUs per instance, xlarge indicates 4 vCPUs per instance, 2xlarge indicates 8 vCPUs per instance, and 3xlarge indicates 12 vCPUs per instance.

The following section describes the columns of the tables in the topics about Intel-, Hygon-, Kunpeng-, and Feiteng-based instance families.

- Instance family: the name of the instance family.
- Processor: the model of the processor supported and recommended for the instance family. For example, Intel V6 represents Intel Cascade Lake processors.

- Virtualization: the virtualization technology used by the instance family, including KVM, first-generation SHENLONG architecture, second-generation SHENLONG architecture, and third-generation SHENLONG architecture.
- Physical machine vCPUs: the number of hyperthreads on an X86-based host or the number of logic cores on an ARM-based host. This number is represented by the number of vCPUs. In the same network environment, the network capability of a single vCPU varies on physical machines that have different numbers of vCPUs. The number of vCPUs is specified for the most recent non-shared instance types to provide a better quality of service (QoS). Unlimited indicates that the number of vCPUs is not specified.
- Network: the network configurations of a host. On the same physical machine, the network capability of a single vCPU varies based on network configurations. Network configurations are specified for the most recent non-shared instance types to provide a better QoS. 10 G indicates that the instance family is used in a 2 × 10 Gbit/s network, and 25 G indicates that the instance family is used in a 2 × 25 Gbit/s network. Unlimited indicates that network configurations are not specified.
- Description: the characteristics of the instance family, such as the number of vCPUs, compute-to-memory ratios, GPUs, and attached disks.
- Remarks: indicates whether the instance family is suitable for new customers. Not recommended indicates that the instance family is not recommended for new customers.

### Resource allocation

Resources are allocated differently to ECS instances of different types.

- For all KVM-based instances, CPU and memory resources must be reserved for virtualization. For example, 8 vCPUs and 32 GiB of memory, 10 vCPUs and 40 GiB of memory, or 10 vCPUs and 48 GiB of memory must be reserved to handle network virtualization, storage virtualization, and system control tasks.
- For all instances that use SHENLONG architecture, no CPU or memory resources need to be reserved for the server. SHENLONG architecture integrates software and hardware and offloads network virtualization, storage virtualization, and system control to a dedicated chip system to achieve zero resource overheads and prevent the performance of virtual machines from fluctuating when demand outstrips resource supplies at the underlying layer.
- Resources of the memory and local disks in hosts cannot be overprovisioned.
- CPU resources of hosts that host shared and burstable instances can be overprovisioned. We recommend that you keep the overprovisioning ratio below 4. CPU resources of hosts that host other instances cannot be overprovisioned.

# 3.2.2. Intel-based instance families

## 3.2.2.1. Shared instance families

Shared instance families are suitable for business scenarios where performance requirements are not high, such as small and medium-sized websites and web applications, development environments, servers, code repositories, microservices, testing and staging environments, lightweight databases, lightweight enterprise applications, and integrated application services. This topic describes the attributes, specifications, and virtualization technologies of shared instance families.

| Instance family | Processor | Virtualization | Physical machine vCPUs | Network | Description | Remarks |
|---|---|---|---|---|---|---|
| ecs.s6-k | Intel V6 | KVM | Unlimited | Unlimited | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, and 64.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.s6-m | Intel V6 | Second-generation SHENLONG architecture | Unlimited | Unlimited | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, and 64.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.s7-k | Intel V7 | KVM | Unlimited | Unlimited | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, and 64.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.anyshare | Intel | KVM | Unlimited | Unlimited | • Allows you to create, modify, and delete custom instance types in the Apsara Uni-manager Operations Console.<br>• Custom instance types are all shared instance types and support only Intel processors. | Not recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.s6 | Intel V6 | Second-generation SHENLONG architecture | Unlimited | Unlimited | • vCPUs: 1, 2, 4, and 8.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:4, and 1:8. | Not recommended |
| ecs.xn4 | Intel V4/V5/V6 | KVM | Unlimited | Unlimited | Provides only the ecs.xn4.small instance type with 1 vCPU and 1 GiB of memory. | Not recommended |
| ecs.n4 | Intel V4/V5/V6 | KVM | Unlimited | Unlimited | • vCPUs: 1, 2, 4, 8, 16, and 32.<br>• CPU-to-memory ratio: 1:2. | Not recommended |
| ecs.mn4 | Intel V4/V5/V6 | KVM | Unlimited | Unlimited | • vCPUs: 1, 2, 4, 8, 16, and 32.<br>• CPU-to-memory ratio: 1:4. | Not recommended |
| ecs.e4 | Intel V4/V5/V6 | KVM | Unlimited | Unlimited | • vCPUs: 1, 2, 4, 8, and 16.<br>• CPU-to-memory ratio: 1:8. | Not recommended |
| ecs.xn4v2 | Intel V4/V5/V6 | KVM | Unlimited | Unlimited | • Provides the ecs.xn4.small instance type with 1 vCPU and 1 GiB of memory.<br>• Supports IPv6 addresses. | Not recommended |
| ecs.n4v2 | Intel V4/V5/V6 | KVM | Unlimited | Unlimited | • vCPUs: 1, 2, 4, 8, 16, and 32.<br>• CPU-to-memory ratio: 1:2.<br>• Supports IPv6 addresses. | Not recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.mn4v2 | Intel V4/V5/V6 | KVM | Unlimited | Unlimited | • vCPUs: 1, 2, 4, 8, 16, and 32.<br>• CPU-to-memory ratio: 1:4.<br>• Supports IPv6 addresses. | Not recommended |
| ecs.e4v2 | Intel V4/V5/V6 | KVM | Unlimited | Unlimited | • vCPUs: 1, 2, 4, 8, and 16.<br>• CPU-to-memory ratio: 1:8.<br>• Supports IPv6 addresses. | Not recommended |

## 3.2.2.2. Dedicated instance families

Dedicated instance families are suitable for business scenarios that require high performance, such as scenarios where large volumes of packets are received and transmitted, web frontend servers, data analysis, batch processing, video encoding, and high-performance scientific and engineering applications. This topic describes the attributes, specifications, and virtualization technologies of dedicated instance families.

| Instance family | Processor | Virtualization | Physical machine vCPUs | Network | Description | Remarks |
|---|---|---|---|---|---|---|
| ecs.g7x-se-x25 | Intel V7 | Third-generation SHENLONG architecture | 128 | 25 G | Supports Shared Block Storage. | Recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.g7x-x25 | Intel V7 | Third-generation SHENLONG architecture | 128 | 25 G | None. | Recommended |
| ecs.g7s-k10 | Intel V7 | KVM | 64 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, and 54.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.g7s-k25 | Intel V7 | KVM | 64 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, and 54.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.g7m-k10 | Intel V7 | KVM | 96 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, and 86.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.g7m-k25 | Intel V7 | KVM | 96 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, and 86.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.g7x-k10 | Intel V7 | KVM | 128 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, and 118.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.g7x-k25 | Intel V7 | KVM | 128 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, and 118.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.g7v-k10 | Intel V7 | KVM | 192 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, and 96.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.g7v-k25 | Intel V7 | KVM | 192 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, and 96.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.g6s-m10 | Intel V6 | Second-generation SHENLONG architecture | 64 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, and 64.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Not recommended |
| ecs.g6s-m25 | Intel V6 | Second-generation SHENLONG architecture | 64 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, and 64.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Not recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.g6m-m10 | Intel V6 | Second-generation SHENLONG architecture | 96 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, and 96.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Not recommended |
| ecs.g6m-m25 | Intel V6 | Second-generation SHENLONG architecture | 96 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, and 96.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Not recommended |
| ecs.g6s-k10 | Intel V6 | KVM | 64 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, and 56.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.g6s-k25 | Intel V6 | KVM | 64 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, and 56.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.g6m-k10 | Intel V6 | KVM | 96 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, and 88.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.g6m-k25 | Intel V6 | KVM | 96 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, and 88.<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.c6 | Intel V6 | Second-generation SHENLONG architecture | 104 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 52, 64, and 104.<br>• CPU-to-memory ratio: 1:2. | Recommended |
| ecs.g6 | Intel V6 | Second-generation SHENLONG architecture | 104 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 52, 64, and 104.<br>• CPU-to-memory ratio: 1:4. | Recommended |
| ecs.r6 | Intel V6 | Second-generation SHENLONG architecture | 104 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 52, 64, and 104.<br>• CPU-to-memory ratio: 1:8. | Recommended |
| ecs.re6 | Intel V6 | Second-generation SHENLONG architecture | 208 | Unlimited | Offers 3 TB of memory and a CPU-to-memory ratio of 1:15. | Recommended |
| ecs.sn1 | Intel V5/V6 | KVM | Unlimited | Unlimited | • vCPUs: 2, 4, 8, 16, 32, and 56.<br>• CPU-to-memory ratio: 1:2. | Not recommended |
| ecs.sn2 | Intel V5/V6 | KVM | Unlimited | Unlimited | • vCPUs: 2, 4, 8, 16, 32, and 56.<br>• CPU-to-memory ratio: 1:4. | Not recommended |
| ecs.se1 | Intel V5/V6 | KVM | Unlimited | Unlimited | • vCPUs: 2, 4, 8, 16, 32, and 56.<br>• CPU-to-memory ratio: 1:8. | Not recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.sn1ne | Intel V5/V6 | KVM | Unlimited | Unlimited | • vCPUs: 2, 4, 8, 12, 16, 24, 32, and 88.<br>• CPU-to-memory ratio: 1:2.<br>• Supports IPv6 addresses. | Not recommended |
| ecs.sn2ne | Intel V5/V6 | KVM | Unlimited | Unlimited | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 56, and 88.<br>• CPU-to-memory ratio: 1:4.<br>• Supports IPv6 addresses. | Not recommended |
| ecs.se1ne | Intel V5/V6 | KVM | Unlimited | Unlimited | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 56, and 88.<br>• CPU-to-memory ratio: 1:8.<br>• Supports IPv6 addresses. | Not recommended |
| ecs.c5 | Intel V5 | Second-generation SHENLONG architecture | 104 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 64, 86, and 88.<br>• CPU-to-memory ratio: 1:2. | Not recommended |
| ecs.g5 | Intel V5 | Second-generation SHENLONG architecture | 104 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 64, 86, and 88.<br>• CPU-to-memory ratio: 1:4. | Not recommended |
| ecs.r5 | Intel V5 | Second-generation SHENLONG architecture | 104 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 64, 86, and 88.<br>• CPU-to-memory ratio: 1:8. | Not recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.re5 | Intel V5 | KVM | 192 | Unlimited | Offers 3 TB of memory and a CPU-to-memory ratio: 1:16. | Not recommended |

## 3.2.2.3. ECS Bare Metal Instance families

**Elastic Compute Service (ECS) Bare Metal Instance families provide robust compute, storage, and network configurations for business scenarios that require dedicated resources, security isolation, and high performance, such as containers, databases, core business of enterprises, and big data computing. This topic describes the attributes, specifications, and virtualization technologies of ECS Bare Metal Instance families.**

| Instance family | Processor | Virtualization | Physical machine vCPUs | Network | Description | Remarks |
|---|---|---|---|---|---|---|
| ecs.ebmg7s-se-x25-c1m8 | Intel V7 | Third-generation SHENLONG architecture | 64 | 25 G | Provides the ecs.ebmg7s-se-x25-c1m8.16xlarge instance type with 64 vCPUs and 512 GiB of memory.<br>Supports Shared Block Storage and allows you to enable NUMA mode. | Recommended |
| ecs.ebmg7m-se-x25-c1m8 | Intel V7 | Third-generation SHENLONG architecture | 96 | 25 G | Provides the ecs.ebmg7m-se-x25-c1m8.24xlarge instance type with 96 vCPUs and 768 GiB of memory.<br>Supports Shared Block Storage and allows you to enable NUMA mode. | Recommended |
| ecs.ebmg7x-se-x25-c1m8 | Intel V7 | Third-generation SHENLONG architecture | 128 | 25 G | Provides the ecs.ebmg7x-se-x25-c1m8.32xlarge instance type with 128 vCPUs and 1,024 GiB of memory.<br>Supports Shared Block Storage and allows you to enable NUMA mode. | Recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.ebmg7s-se-numaoff-x25-c1m8 | Intel V7 | Third-generation SHENLONG architecture | 64 | 25 G | Provides the ecs.ebmg7s-se-numaoff-x25-c1m8.16xlarge instance type with 64 vCPUs and 512 GiB of memory.<br>Supports Shared Block Storage and allows you to disable NUMA mode. | Not recommended |
| ecs.ebmg7m-se-numaoff-x25-c1m8 | Intel V7 | Third-generation SHENLONG architecture | 96 | 25 G | Provides the ecs.ebmg7m-se-numaoff-x25-c1m8.24xlarge instance type with 96 vCPUs and 768 GiB of memory.<br>Supports Shared Block Storage and allows you to disable NUMA mode. | Not recommended |
| ecs.ebmg7x-se-numaoff-x25-c1m8 | Intel V7 | Third-generation SHENLONG architecture | 128 | 25 G | Provides the ecs.ebmg7x-se-numaoff-x25-c1m8.32xlarge instance type with 128 vCPUs and 1,024 GiB of memory.<br>Supports Shared Block Storage and allows you to disable NUMA mode. | Not recommended |
| ecs.ebmg7s-x25-c1m8 | Intel V7 | Third-generation SHENLONG architecture | 64 | 25 G | Provides the ecs.ebmg7s-x25-c1m8.16xlarge instance type with 64 vCPUs and 512 GiB of memory and allows you to enable NUMA mode. | Recommended |
| ecs.ebmg7m-x25-c1m8 | Intel V7 | Third-generation SHENLONG architecture | 96 | 25 G | Provides the ecs.ebmg7m-x25-c1m8.24xlarge instance type with 96 vCPUs and 768 GiB of memory and allows you to enable NUMA mode. | Recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.ebmg7x-x25-c1m8 | Intel V7 | Third-generation SHENLONG architecture | 128 | 25 G | Provides the ecs.ebmg7x-x25-c1m8.32xlarge instance type with 128 vCPUs and 1,024 GiB of memory and allows you to enable NUMA mode. | Recommended |
| ecs.ebmg6s-m10-c1m4 | Intel V6 | Second-generation SHENLONG architecture | 64 | 10 G | Provides the ecs.ebmg6s-m10-c1m4.16xlarge instance type with 64 vCPUs and 256 GiB of memory. | Recommended |
| ecs.ebmg6s-m25-c1m4 | Intel V6 | Second-generation SHENLONG architecture | 64 | 25 G | Provides the ecs.ebmg6s-m25-c1m4.16xlarge instance type with 64 vCPUs and 256 GiB of memory. | Recommended |
| ecs.ebmg6s-m10-c1m24 | Intel V6 | Second-generation SHENLONG architecture | 64 | 10 G | Provides the ecs.ebmg6s-m10-c1m24.16xlarge instance type with 64 vCPUs and 1,536 GiB of memory. | Recommended |
| ecs.ebmg6s-m25-c1m24 | Intel V6 | Second-generation SHENLONG architecture | 64 | 25 G | Provides the ecs.ebmg6s-m25-c1m24.16xlarge instance type with 64 vCPUs and 1,536 GiB of memory. | Recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.ebmg6m-m10-c1m4 | Intel V6 | Second-generation SHENLONG architecture | 96 | 10 G | Provides the ecs.ebmg6m-m10-c1m4.24xlarge instance type with 96 vCPUs and 384 GiB of memory. | Recommended |
| ecs.ebmg6m-m25-c1m4 | Intel V6 | Second-generation SHENLONG architecture | 96 | 10 G | Provides the ecs.ebmg6m-m25-c1m4.24xlarge with 96 vCPUs and 384 GiB of memory. | Recommended |
| ecs.ebmre6-3t | Intel V6 | Second-generation SHENLONG architecture | 208 | 25 G | Provides the ecs.ebmre6-3t.52xlarge with 208 vCPUs and 3,072 GiB of memory. | Recommended |
| ecs.ebmc6 | Intel V6 | Second-generation SHENLONG architecture | 104 | 25 G | Provides the ecs.ebmc6.26xlarge instance type with 104 vCPUs and 192 GiB of memory. | Recommended |
| ecs.ebmg6 | Intel V6 | Second-generation SHENLONG architecture | 104 | 25 G | Provides the ecs.ebmg6.26xlarge instance type with 104 vCPUs and 384 GiB of memory. | Recommended |
| ecs.ebmr6 | Intel V6 | Second-generation SHENLONG architecture | 104 | 25 G | Provides the ecs.ebmr6.26xlarge instance type with 104 vCPUs and 768 GiB of memory. | Recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.ebmre6p | Intel V6 | Second-generation SHENLONG architecture | 104 | 25 G | Uses 1,536 GiB of Intel Optane persistent memory (AEP). | Recommended |
| ecs.ebmg5 | Intel V5 | First-generation SHENLONG architecture | 96 | 10 G | Provides the ecs.ebmg5.24xlarge instance type with 96 vCPUs and 384 GiB of memory. | Not recommended |
| ecs.ebmg5s | Intel V5 | Second-generation SHENLONG architecture | 96 | 25 G | Provides the ecs.ebmg5s.24xlarge instance type with 96 vCPUs and 384 GiB of memory. | Not recommended |

# 3.2.2.4. Heterogeneous computing and heterogeneous ECS Bare Metal Instance families

Heterogeneous computing instance families and heterogeneous ECS Bare Metal Instance families use various accelerators including NVIDIA T4 GPUs, NVIDIA V100 GPUs, and FPGAs, and are suitable for business scenarios such as AI inference, computer vision, speech recognition, speech synthesis, machine translation, recommendation systems, real-time rendering, deep learning, and scientific computing applications. This topic describes the attributes, specifications, and virtualization technologies of heterogeneous computing instance families and heterogeneous ECS Bare Metal Instance families.

| Instance family | Processor | Virtualization | vCPUs of physical server | Network | Description | Remarks |
|---|---|---|---|---|---|---|
| ecs.gn6i | Intel V5 and V6 | KVM | Unlimited | Unlimited | Uses one to eight NVIDIA T4 GPUs and offers a CPU-to-memory ratio of 1:4 | Recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.gn6v | Intel V5 and V6 | KVM | Unlimited | Unlimited | Uses one to eight NVIDIA V100 GPUs and offers a CPU-to-memory ratio of 1:4 | Recommended |
| ecs.vgn6i | Intel V5 and V6 | Second-generation SHENLONG architecture | Unlimited | Unlimited | Uses vGPUs<br>Supports the 1/4 and 1/2 compute capacity of NVIDIA Tesla T4 GPUs | Recommended |
| ecs.ebmgn6i | Intel V5 and V6 | Second-generation SHENLONG architecture | Unlimited | Unlimited | ECS Bare Metal Instance family that uses four NVIDIA T4 GPUs | Recommended |
| ecs.ebmgn6i-2 | Intel V5 and V6 | Second-generation SHENLONG architecture | Unlimited | Unlimited | ECS Bare Metal Instance family that uses two NVIDIA T4 GPUs | Recommended |
| ecs.ebmgn6i-8 | Intel V5 and V6 | Second-generation SHENLONG architecture | Unlimited | Unlimited | ECS Bare Metal Instance family that uses eight NVIDIA T4 GPUs | Recommended |
| ecs.ebmgn6e-2 | Intel V6 | Second-generation SHENLONG architecture | Unlimited | Unlimited | ECS Bare Metal Instance family that uses two NVIDIA V100 GPUs | Recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.gn3 | Intel V4 | KVM | Unlimited | Unlimited | Uses one to two NVIDIA K2 GPUs | Not recommended |
| ecs.gn4 | Intel V4 | KVM | Unlimited | Unlimited | Uses one to two NVIDIA M40 GPUs | Not recommended |
| ecs.gn5 | Intel V4 | KVM | Unlimited | Unlimited | Uses one to eight NVIDIA P100 GPUs | Not recommended |
| ecs.gn5e | Intel V5 | KVM | Unlimited | Unlimited | Uses one to eight NVIDIA P4 GPUs and offers a CPU-to-memory ratio of 1:6 | Not recommended |
| ecs.gn5i | Intel V4 | KVM | Unlimited | Unlimited | Uses one to eight NVIDIA P4 GPUs and offers a CPU-to-memory ratio of 1:4 | Not recommended |
| ecs.gn5t | Intel V4 | KVM | Unlimited | Unlimited | Uses two to eight NVIDIA 1080Ti GPUs and offers a CPU-to-memory ratio of 1:4 | Not recommended |

# 3.2.2.5. SCC instance families

Super Computing Cluster (SCC) instance families provide computing cluster services with ultimate computing performance and parallel efficiency by using the high-speed InfiniBand network on top of Elastic Compute Service (ECS) Bare Metal Instance families. SCC instance families are suitable for scenarios such as high-performance computing, artificial intelligence, machine learning, scientific and engineering computing, data analysis, and audio and video processing. This topic describes the attributes, specifications, and virtualization technologies of SCC instance families.

| Instance family | Processor | Virtualization | Physical machine vCPUs | Network | Description | Remarks |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.sccgn6e | Intel V6 | Second-generation SHENLONG architecture | 104 | Unlimited | Uses eight NVIDIA V100 GPUs and a 100 Gbit/s InfiniBand network. | Recommended |
| ecs.sccgn6p | Intel V5/V6 | Second-generation SHENLONG architecture | 96 | Unlimited | Uses eight NVIDIA V100 GPUs and a 100 Gbit/s InfiniBand network. | Recommended |
| ecs.scch5 | Intel V5 | Second-generation SHENLONG architecture | 64 | Unlimited | Uses processors with high clock speeds that deliver a base frequency of 3.1 GHz and an all-core turbo frequency of 3.5 GHz. | Not recommended |
| ecs.sccg5ib | Intel V5 | First-generation SHENLONG architecture | 96 | 10 G | Uses a 100 Gbit/s InfiniBand network. | Not recommended |

# 3.2.2.6. Instance families equipped with local HDDs

Instance families equipped with local HDDs are suitable for business scenarios such as mass storage and offline computing that have high requirements for big data computing, storage, and analysis and can meet the high requirements of distributed computing services such as Hadoop in terms of storage performance, storage capacity, and internal bandwidth. This topic describes the attributes, specifications, and virtualization technologies of instance families equipped with local HDDs.

| Instance family | Processor | Virtualization | Physical machine vCPUs | Network | Description | Remarks |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.d7s-k10-8t | Intel V7 | KVM | 64 | 10 G | Uses twelve 8-TB SATA HDDs | Recommended |
| ecs.d7s-k25-8t | Intel V7 | KVM | 64 | 25 G | Uses twelve 8-TB SATA HDDs. | Recommended |
| ecs.d7m-k10-8t | Intel V7 | KVM | 96 | 10 G | Uses twelve 8-TB SATA HDDs. | Recommended |
| ecs.d7m-k25-8t | Intel V7 | KVM | 96 | 25 G | Uses twelve 8-TB SATA HDDs. | Recommended |
| ecs.d7x-k10-12t | Intel V7 | KVM | 128 | 10 G | Uses twelve 12-TB SATA HDDs. | Recommended |
| ecs.d7x-k25-12t | Intel V7 | KVM | 128 | 25 G | Uses twelve 12-TB SATA HDDs. | Recommended |
| ecs.d2-zyy | Intel V5/V6 | KVM | Unlimited | Unlimited | Uses twelve 8-TB SATA HDDs. | Recommended |

# 3.2.2.7. Burstable instance families

Burstable instance families provide baseline CPU performance and are burstable but limited by accrued CPU credits. They are suitable for business scenarios that do not require high performance of virtual machines, such as web applications and small and medium-sized websites. This topic describes the attributes, specifications, and virtualization technologies of burstable instance families.

| Instance family | Processor | Virtualization | vCPUs of physical server | Network | Description | Remarks |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.t5 | Intel V5 and V6 | KVM | Unlimited | Unlimited | Provides baseline CPU performance and is burstable but limited by accrued CPU credits. | Not recommended. Shared instance families are recommended. |

# 3.2.3. Hygon-based instance families

## 3.2.3.1. Shared instance families

Shared instance families are suitable for business scenarios that do not require high performance of virtual machines, such as small and medium-sized websites and web applications, development environments, servers, code repositories, microservices, testing and staging environments, lightweight databases, lightweight enterprise applications, and integrated application services. This topic describes the attributes, specifications, and virtualization technologies of shared instance families.

| Instance family | Processor | Virtualization | vCPUs of physical server | Network | Description | Remarks |
|---|---|---|---|---|---|---|
| ecs.s6-hg-k | Hygon 2 | KVM | Unlimited | Unlimited | vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, and 64 | Recommended |
| ecs.ghg-s | Hygon 1 and Hygon 2 | KVM | Unlimited | Unlimited | CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Not recommended |

## 3.2.3.2. Dedicated instance families

Dedicated instance families are suitable for business scenarios that require high performance, such as scenarios where large volumes of packets are received and transmitted, web frontend servers, data analysis, batch processing, video encoding, and high-performance scientific and engineering applications. This topic describes the attributes, specifications, and virtualization technologies of dedicated instance families.

| Instance family | Processor | Virtualization | Physical machine vCPUs | Network | Description | Remarks |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.g6s-hg-k10 | Hygon 2 | KVM | 64 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, and 56<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8. | Recommended |
| ecs.g6s-hg-k25 | Hygon 2 | KVM | 64 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, and 56<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.g6x-hg-k10 | Hygon 2 | KVM | 128 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, and 120<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.g6x-hg-k25 | Hygon 2 | KVM | 128 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, and 120<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.ghg | Hygon 1 and Hygon 2 | KVM | Unlimited | Unlimited | CPU-to-memory ratio: 1:4 | Not recommended |

# 3.2.3.3. Heterogeneous computing instance families

Heterogeneous computing instance families use various accelerators including NVIDIA T4 GPUs, NVIDIA V100 GPUs, and FPGAs, and are suitable for business scenarios such as AI inference, computer vision, speech recognition, speech synthesis, machine translation, recommendation systems, real-time rendering, deep learning, and scientific computing applications. This topic describes the attributes, specifications, and virtualization technologies of heterogeneous computing instance families.

| Instance family | Processor | Virtualization | vCPUs of physical server | Network | Description | Remarks |
|---|---|---|---|---|---|---|
| ecs.gn6ih | Hygon 2 | KVM | 128 vCPUs | Unlimited | Uses eight NVIDIA T4 GPUs and offers a CPU-to-memory ratio of 1:4 | Recommended |
| ecs.gn6hv | Hygon 1 | KVM | 128 vCPUs | Unlimited | Uses one to two NVIDIA V100 GPUs and offers a CPU-to-memory ratio of 1:4 | Recommended |

## 3.2.3.4. Instance families equipped with local HDDs

Instance families equipped with local HDDs are suitable for business scenarios that have high requirements for big data computing, storage, and analysis, such as mass storage and offline computing scenarios, and can also meet the high requirements of distributed computing services such as Hadoop in terms of storage performance, storage capacity, and internal bandwidth. This topic describes the attributes, specifications, and virtualization technologies of instance families equipped with local HDDs.

| Instance family | Processor | Virtualization | vCPUs of physical server | Network | Description | Remarks |
|---|---|---|---|---|---|---|
| ecs.dhg | Hygon 1 and Hygon 2 | KVM | Unlimited | Unlimited | Uses twelve 8-TB SATA HDDs | Recommended |

# 3.2.4. Kunpeng-based instance families

## 3.2.4.1. Shared instance families

Shared instance families are suitable for business scenarios that do not require high performance of virtual machines, such as small and medium-sized websites and web applications, development environments, servers, code repositories, microservices, testing and staging environments, lightweight databases, lightweight enterprise applications, and integrated application services. This topic describes the attributes, specifications, and virtualization technologies of shared instance families.

| Instance family | Processor | Virtualization | vCPUs of physical server | Network | Description | Remarks |
|---|---|---|---|---|---|---|
| ecs.s5-kp-k | Kunpeng920 | KVM | Unlimited | Unlimited | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, and 64<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.gkp-s | Kunpeng920 | KVM | Unlimited | Unlimited | CPU-to-memory ratio: 1:4 | Not recommended |

## 3.2.4.2. Dedicated instance families

Dedicated instance families are suitable for business scenarios that require high performance, such as scenarios where large volumes of packets are received and transmitted, web frontend servers, data analysis, batch processing, video encoding, and high-performance scientific and engineering applications. This topic describes the attributes, specifications, and virtualization technologies of dedicated instance families.

| Instance family | Processor | Virtualization | Physical machine vCPUs | Network | Description | Remarks |
|---|---|---|---|---|---|---|
| ecs.g5s-kp-k10 | Kunpeng920 | KVM | 64 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, and 56<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.g5s-kp-k25 | Kunpeng920 | KVM | 64 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, and 56<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.g5m-kp-k10 | Kunpeng920 | KVM | 96 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, and 88<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.g5m-kp-k25 | Kunpeng920 | KVM | 96 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, and 88<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.g5x-kp-k10 | Kunpeng920 | KVM | 128 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, and 120<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.g5x-kp-k25 | Kunpeng920 | KVM | 128 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, and 120<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.g5m-kp-m10 | Kunpeng920 | Second-generation SHENLONG architecture | 96 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, and 96<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.g5m-kp-m25 | Kunpeng920 | Second-generation SHENLONG architecture | 96 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, and 96<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.gkp | Kunpeng920 | KVM | Unlimited | Unlimited | CPU-to-memory ratio: 1:4 | Not recommended |
| ecs.gkp-m | Kunpeng920 | Second-generation SHENLONG architecture | Unlimited | Unlimited | CPU-to-memory ratio: 1:4 | Not recommended |

## 3.2.4.3. Instance families equipped with local HDDs

Instance families equipped with local HDDs are suitable for business scenarios that have high requirements for big data computing, storage, and analysis, such as mass storage and offline computing scenarios, and can also meet the high requirements of distributed computing services such as Hadoop in terms of storage performance, storage capacity, and internal bandwidth. This topic describes the attributes, specifications, and virtualization technologies of instance families equipped with local HDDs.

| Instance family | Processor | Virtualization | vCPUs of physical server | Network | Description | Remarks |
|---|---|---|---|---|---|---|
| ecs.dkp | Kunpeng920 | KVM | Unlimited | Unlimited | Uses twelve 8-TB SATA HDDs | Recommended |

# 3.2.5. Feiteng-based instance families

## 3.2.5.1. Shared instance families

Shared instance families are suitable for business scenarios that do not require high performance of virtual machines, such as small and medium-sized websites and web applications, development environments, servers, code repositories, microservices, testing and staging environments, lightweight databases, lightweight enterprise applications, and integrated application services. This topic describes the attributes, specifications, and virtualization technologies of shared instance families.

| Instance family | Processor | Virtualization | vCPUs of physical server | Network | Description | Remarks |
|---|---|---|---|---|---|---|
| ecs.s5-ft-k | FT-2000+ | KVM | Unlimited | Unlimited | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, and 64<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.s6-ft-k | FT-S2500 | KVM | Unlimited | Unlimited | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, and 64<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |

## 3.2.5.2. Dedicated instance families

Dedicated instance families are suitable for business scenarios that require high performance, such as scenarios where large volumes of packets are received and transmitted, web frontend servers, data analysis, batch processing, video encoding, and high-performance scientific and engineering applications. This topic describes the attributes, specifications, and virtualization technologies of dedicated instance families.

| Instance family | Processor | Virtualization | Physical machine vCPUs | Network | Description | Remarks |
|---|---|---|---|---|---|---|
| ecs.g5s-ft-k10 | FT-2000+ | KVM | 64 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, and 56<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |

| | | | | | | |
|---|---|---|---|---|---|---|
| ecs.g5s-ft-k25 | FT-2000+ | KVM | 64 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, and 56<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.g6x-ft-k10 | FT-S2500 | KVM | 128 | 10 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, and 120<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.g6x-ft-k25 | FT-S2500 | KVM | 128 | 25 G | • vCPUs: 2, 4, 8, 12, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, and 120<br>• CPU-to-memory ratios: 1:1, 1:2, 1:3, 1:4, 1:6, and 1:8 | Recommended |
| ecs.gft | FT-2000+ | KVM | Unlimited | Unlimited | CPU-to-memory ratio: 1:4 | Not recommended |

# 4.Auto Scaling (ESS)

## 4.1. Product Introduction

### 4.1.1. What is Auto Scaling?

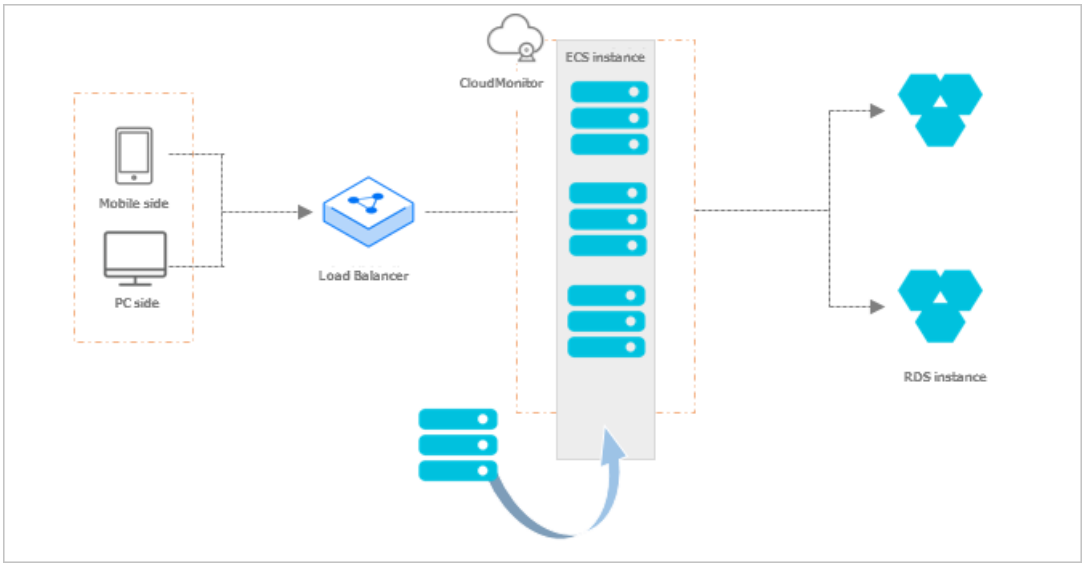
Auto Scaling automatically adjusts your elastic computing resources based on your business requirements and policies that you define.

When demand for services spikes, Auto Scaling automatically scales out Elastic Compute Service (ECS) instances based on your configurations to maintain sufficient computing resources. When demand for services drops, Auto Scaling automatically scales in ECS instances to save costs.

Auto Scaling provides the following features:

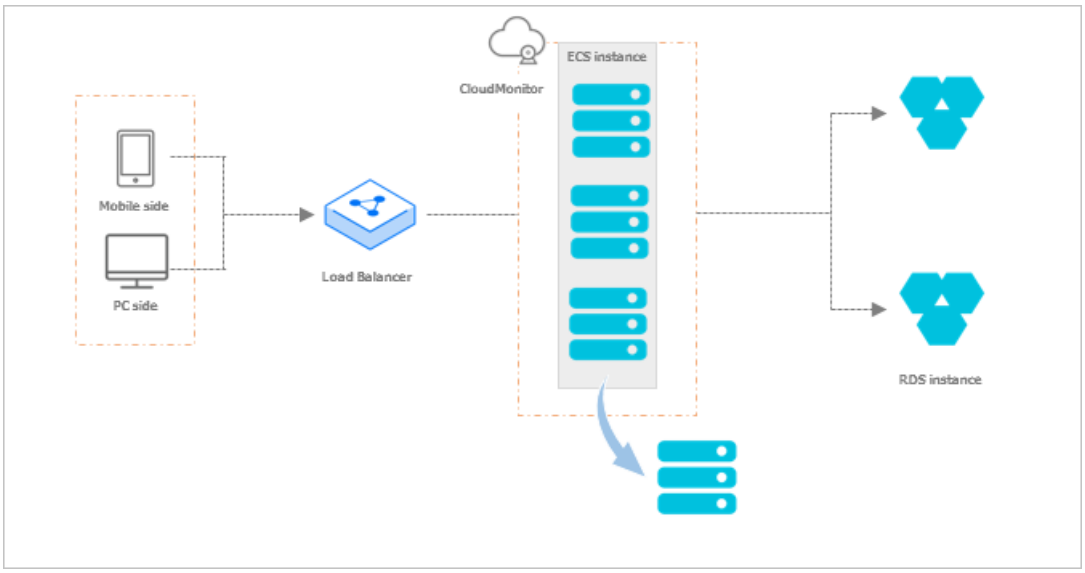
- Scale-out

  When demand for services suddenly grows, Auto Scaling automatically scales out the underlying resources. This ensures that resources are not overloaded and maintains the responsiveness of your servers. For example, if the vCPU utilization of ECS instances exceeds 80%, Auto Scaling scales out ECS resources based on your configurations. During the scale-out event, Auto Scaling automatically creates ECS instances, adds the ECS instances to a scaling group, and then adds the new instances to the backend server groups of the associated Server Load Balancer (SLB) instances and the whitelists of the associated ApsaraDB RDS instances. The following figure shows how a scale-out event is implemented.



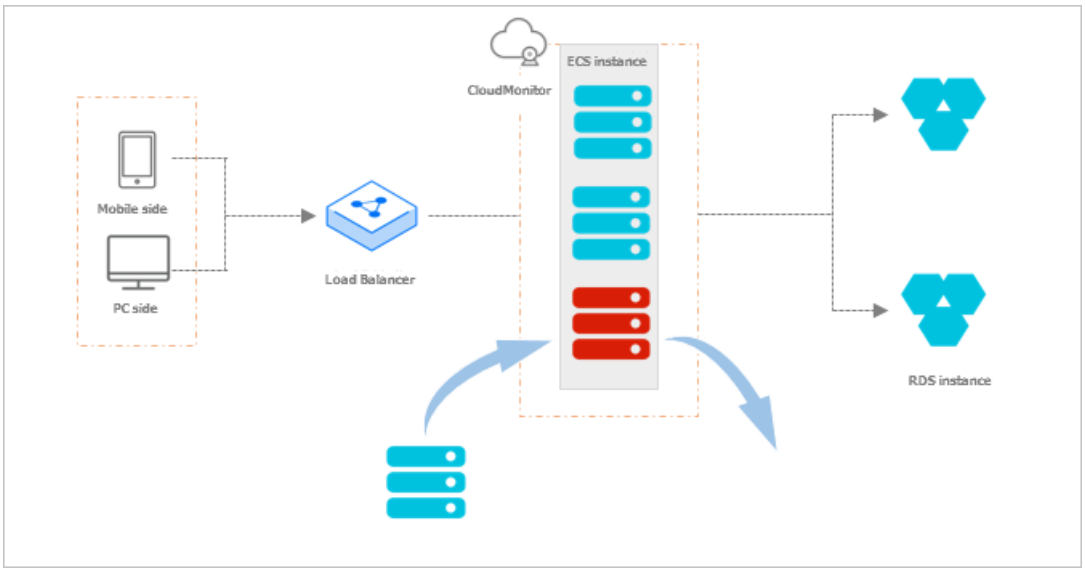

- Scale-in

When demand for services drops, Auto Scaling automatically releases underlying resources to prevent waste of resources and reduce costs. For example, if the vCPU utilization of ECS instances in a scaling group falls below 30%, Auto Scaling automatically scales in ECS instances based on your configurations. 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 and the whitelists of the associated ApsaraDB RDS instances. The following figure shows how a scale-in event is implemented.




- 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. Elastic recovery 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 how an elastic recovery event is implemented.




# 4.1.2. 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.

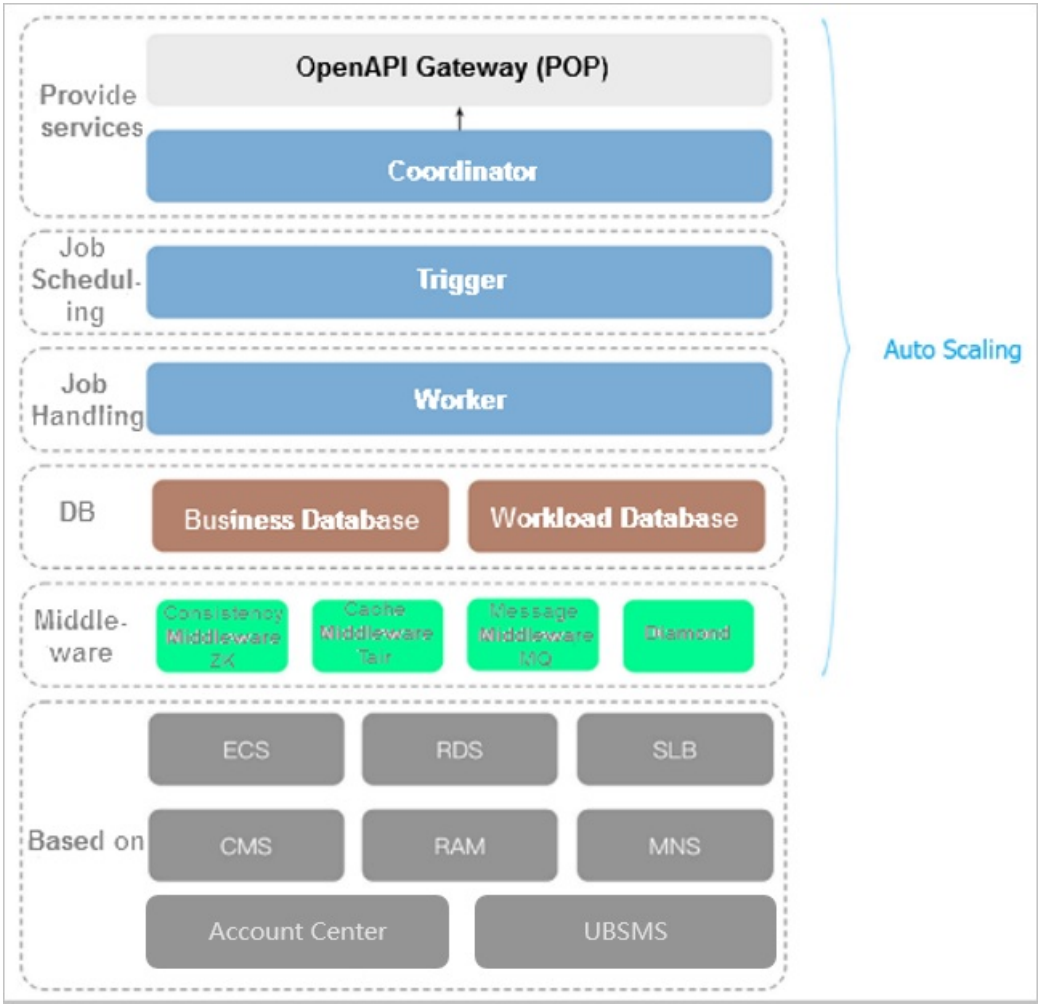
- Various scaling modes for you to mix and match

  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.

# 4.1.3. 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. |

| Component | Description |
| --- | --- |
| 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. |

# 4.1.4. Features

## 4.1.4.1. Scaling groups

A scaling group is a group of Elastic Compute Service (ECS) instances that are dynamically scaled based on the scenario. You can specify the minimum or maximum number of ECS instances in a scaling group. You can also specify the Server Load Balancer (SLB) and ApsaraDB RDS instances that are associated with the scaling group. Auto Scaling can automatically adjust the number of ECS instances in a scaling group. You can also manually adjust the number of ECS instances in a scaling group.

### Maximum and minimum number of instances

Maximum Number of Instances: the maximum number of ECS instances that a scaling group can contain. Set this parameter based on your business requirements to control costs.

Minimum Number of Instances: the minimum number of ECS instances that a scaling group can contain. Set this parameter based on your business requirements to ensure that daily business requirements can be met.

### Associate SLB instances

After an SLB instance is associated with a scaling group, ECS instances that are added to the scaling group are automatically added as backend servers of the SLB instance. The SLB instance then forwards requests to the ECS instances. The ECS instances removed from the scaling group are automatically removed from the backend servers of the SLB instance. You can specify a server group to which to add the ECS instances. ECS instances can be added to the following types of server groups:

- Default server group: the group of ECS instances that are used to receive requests. If a listener is not associated with a vServer group or a primary/secondary server group, requests are forwarded to the ECS instances in the default server group.
- vServer group: If you want to forward different requests to different backend servers or configure domain name- or URL-based routing methods, you can use vServer groups.

To associate a scaling group with SLB instances, make sure that the following requirements are met:

- You have one or more SLB instances in the **Running** state. For more information, see *SLB User G uide*.

- SLB instances and the scaling group to be associated must be in the same region, the same organization, and the same resource set.
- The SLB instance and the scaling group are in the same virtual private cloud (VPC) network if their network type is VPC.
- If the network type of the SLB instance is classic network, the network type of the scaling group is VPC, and the backend server group of the SLB instance contains VPC-type ECS instances, the ECS instances and the scaling group must be in the same VPC.
- You must configure at least one listener for an SLB instance. For more information, see *SLB User Guide*.
- You must enable health checks for your SLB instances. For more information, see *SLB User Guide*.

### Associate ApsaraDB RDS instances

After an ApsaraDB RDS instance is associated with a scaling group, the internal IP addresses of the ECS instances that are added to the scaling group are automatically added to the whitelist of the ApsaraDB RDS instance. Then, the ECS instances and the ApsaraDB instance can communicate with each other over the internal network. The whitelist of the ApsaraDB RDS instance does not include the internal IP addresses of the ECS instances that are removed from the scaling group.

To associate ApsaraDB RDS instances with a scaling group, make sure that the following requirements are met:

- You have one or more ApsaraDB RDS instances in the **Running** state. For more information, see *ApsaraDB RDS Product Introduction*.
- The ApsaraDB RDS instances and the scaling group to be associated must be in the same region, the same organization, and the same resource set.

### Health checks for ECS instances

This feature is enabled by default and cannot be disabled. After this feature is enabled, Auto Scaling regularly checks the status of ECS instances in the scaling group. If an ECS instance in a scaling group is not in the Running state, Auto Scaling considers the instance to be unhealthy and automatically removes the instance. The removal method is based on how the ECS instance is added:

- If an ECS instance is automatically created, Auto Scaling immediately removes and releases it.
- If an ECS instance is manually added, Auto Scaling immediately removes it, but does not stop or release it.

If the removal of unhealthy instances results in the number of instances to be less than the minimum number, Auto Scaling automatically creates the required number of ECS instances to maintain the minimum number. Health checks for ECS instances can ensure 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.

## 4.1.4.2. Scaling configurations

A scaling configuration defines the configuration of ECS instances used for automatic scaling. During a scale-out event, Auto Scaling creates ECS instances for a scaling group based on the scaling configuration of the scaling group and then adds these ECS instances to the scaling group.

A scaling configuration specifies the region, zone, security group, instance type, image, and storage information of the ECS instances in a scaling group. Scale-out events can be triggered by using scheduled or event-triggered tasks. When a scale-out event is triggered, Auto Scaling uses the scaling configuration as a template to automatically create ECS instances.

You must specify a security group when you create a scaling configuration. Make sure that the security group and the scaling group are in the same virtual private cloud (VPC). If no security group exists in the VPC of the scaling group, you must create a security group. For more information, see *ECS User Guide*.

Only one scaling configuration can be in the Enabled state in a scaling group. After one scaling configuration in a scaling group is applied, the other scaling configurations enter the Disabled state.

## 4.1.4.3. Scaling rules

A scaling rule defines the specific operations when a scaling group is scaled out or in, such as adding or removing ECS instances.

Scaling rules allow you to add a specified number of ECS instances to a scaling group or remove a specified number of ECS instances from a scaling group.

Scaling rules can be executed automatically by using scheduled tasks or event-triggered tasks. You can also manually execute scaling rules. After a scaling rule is executed, the number of ECS instances in the scaling group may be outside the specified range. In this case, Auto Scaling automatically adjusts the number of ECS instances to ensure that the number of ECS instances in the scaling group stays within the specified range.

## 4.1.4.4. Scheduled tasks

Auto Scaling allows you to preconfigure scheduled tasks based on historical peak times. A scheduled task is preconfigured to execute the specified scaling rule at the specified time. At the specified time, the scheduled task automatically scales computing resources. Scheduled tasks help you operate your business smoothly and reduce your costs.

You can preconfigure a scheduled task to run based on historical peak and off-peak workloads. When you preconfigure a scheduled task, you can specify to execute the task before and after the peak times. You can also configure both scale-out and scale-in rules for the task. The scheduled task triggers the scale-out rule before the specified time to prepare sufficient computing resources. The scheduled task triggers the scale-in rule after the specified time to release idle computing resources.

Scheduled tasks also support the following features:

- You can specify a recurrence period and expiration time for a scheduled task. Before a scheduled task expires, Auto Scaling executes this task on a daily, weekly, or monthly basis. Scheduled tasks execute flexible rules to respond to periodic business changes.
- Scheduled tasks can be automatically retried at a specified time period. If a scaling task cannot be executed as expected, Auto Scaling retries this task at a specified time period to prevent the overall result of this task from being affected by a single failure.
- If multiple scheduled tasks need to be executed within the same minute, Auto Scaling executes the most recently created scheduled task.

# 4.1.4.5. Event-triggered tasks

Event-triggered tasks can be used based on Auto Scaling and CloudMonitor to dynamically manage scaling groups. After you create and enable an event-triggered task, Auto Scaling collects data for the specified metric in real time and triggers an alert when the specified condition is met. Then, Auto Scaling executes the corresponding scaling rule to scale ECS instances in the scaling group.

## Overview

A scheduled task is preconfigured to execute the specified scaling rule at the specified time. You can create scheduled tasks based on predictable business changes. However, when sudden traffic changes occur, scheduled tasks may fail your expectations. You can use event-triggered tasks to trigger scaling rules in a more flexible manner. Auto Scaling can add instances to a scaling group during peak hours, and release instances during off-peak hours to help you save costs.

Event-triggered tasks use CloudMonitor to monitor specific metrics and collect metric values in real time. When the metric values meet the alert conditions, alerts are triggered to execute specified scaling rules. You can use event-triggered tasks to adjust the number of instances in a scaling group based on business changes. This can ensure that the values of monitoring metrics are within your expected range.

## Limits

During the cooldown period, Auto Scaling does not implement the scaling rule that is triggered by an event-triggered task. In most cases, it takes a few minutes for Auto Scaling to add ECS instances to a scaling group, start the instances, deploy services and collect monitoring metrics. We recommend that you specify an appropriate cooldown period based on your business requirements. This can ensure that the scaling rule is not repeatedly triggered when the monitoring metrics of newly added instances are not collected.

# 4.1.5. 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.

# 4.1.6. Limits

This topic describes the limits of Auto Scaling.

- Auto Scaling can automatically scale the number of Elastic Compute Service (ECS) instances in a scaling group, but cannot automatically upgrade or downgrade configurations of the ECS instances, such as vCPUs, memory, and bandwidth.

- Applications deployed on the ECS instances in a scaling group must be stateless and horizontally scalable.
- ECS instances in a scaling group can be automatically released. We recommend that you do not store information such as sessions, application data, or logs on the ECS instances in a scaling group. If you need to store data of the applications deployed on the ECS instances, store status information such as sessions on the independent ECS instances, store application data in ApsaraDB RDS, and store logs in Log Service. For more information, see *What is ApsaraDB RDS?* in *ApsaraDB RDS Product Introduction* and *What is Log Service?*in *Log Service Product Introduction*.
- 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 in a scaling group. |
| Scaling rule | You can create a maximum of 50 scaling rules in a scaling group. |
| ECS instance | You can add a maximum of 1,000 ECS instances to a scaling group. |

# 4.1.7. 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.Container Registry

## 5.1. Product Introduction

### 5.1.1. Container Registry

Container Registry is a platform that allows you to manage and distribute cloud-native artifacts in a secure and efficient manner. Cloud-native artifacts include container images and Helm charts that meet the standards of Open Container Initiative (OCI). Container Registry provides the following features: image permission management, synchronous image distribution, and content signing. The features allow you to manage the entire lifecycle of container images. Container Registry simplifies the setup and O&M of container registry. Container Registry is integrated with Alibaba Cloud services such as Container Service for Kubernetes (ACK) to easily create and deliver a one-stop solution for cloud-native applications.

#### 5.1.1.1. Features

Container Registry is a platform that allows you to manage and distribute cloud-native artifacts in a secure and efficient manner. Cloud-native artifacts include container images and Helm charts that meet the standards of Open Container Initiative (OCI). Container Registry provides the following features: artifact management, image replication, artifact security, and deployment integration.

##### Artifact management

- Secure management: You can securely manage container images by namespace.
- Lifecycle management: You can query artifacts and image tags. You can also delete artifacts and image repositories.
- Fine-grained permission management: You can manage user permissions, Apsara Stack Cloud Management (ASCM) departments, and resource sets.
- Version immutability: You can configure version immutability for OCI artifacts.

##### Image replication

- Trigger: If a container image is updated, the corresponding event is automatically triggered.
- Image replication: You can manually trigger the replication of a container image of a specific version to implement geo-disaster recovery for container images. Container Registry can also automatically replicate a container image across multiple accounts after the image is pushed to an image repository.

##### Artifact security

- Encrypted image distribution: You can configure secure HTTPS protocol to distribute container images.
- Container image signing: This feature prevents man-in-the-middle (MITM) attacks and unauthorized image updates or deployments. This ensures image consistency and security from distribution to deployment.
- Image scanning: This feature allows you to scan container images to identify vulnerabilities.

### Deployment integration

- Image pulls without a secret: You can configure image pulls without a secret in the Alibaba Cloud Container Service for Kubernetes (ACK) console. This way, you do not need to specify a secret for each image pull.
- Image selection: You can select image repositories and image tags when you configure a Deployment in the ACK console.

## 5.1.1.2. Benefits

Container Registry provides you with the following benefits: ease of use, security, and integrability.

### Ease of use

- Allows you to create an image repository without the need to manually build and maintain the images.
- Allows you to pull and push images across multiple regions in a fast and stable manner.

### Security and controllability

- Provides an all-in-one management system for image permissions. This ensures secure and convenient image sharing.
- Provides the image scanning feature to identify vulnerabilities in images and prompt vulnerability levels.

### Efficient distribution

- Supports large-scale image distribution in a single region and concurrent image pulls on 500 nodes.
- Supports cross-region image replication and cross-cloud synchronous image distribution. Supports manual and automatic image replication.

### Seamless integration with other Alibaba Cloud services

Integrated with Alibaba Cloud services such as Container Service for Kubernetes (ACK) to implement continuous deployment after images are updated.

## 5.1.1.3. Scenarios

Container Registry is suitable for scenarios, such as DevOps, continuous delivery, and automatic image replication.

### DevOps and continuous delivery

Container Registry is integrated with Jenkins to automate the DevOps pipeline from code committing to application deployments and ensure that code is committed for deployment only after the code passes automated testing. This simplifies application deployments and accelerates application iterations.

- DevOps automation

Automates the DevOps pipeline, from code updates to code builds, image builds, and application deployments.

- Environment consistency

  Allows you to deliver code, and deliver runtime environments that are built based on immutable architectures.

- Continuous feedback

  Returns results in real time after integration or delivery.

## Automatic image replication

If the container business of an enterprise is deployed in multiple regions and multiple clouds, container applications of the enterprise must be deployed across multiple regions and clouds after submission. Container Registry provides the image replication feature to improve automated distribution efficiency and disaster recovery capabilities and reduce manual O&M costs.

- Multi-scenario replication
  - Supports cross-region, cross-cloud, and cross-account image replication.
  - Supports manual replication. Supports automatic replication after images are updated.
- Optimized scheduling

  Optimizes the scheduling of replication to increase the success rate of replication.

- Security compliance

  Supports encryption of replication links to ensure the security of replicated data.

# 6.Container Service for Kubernetes

## 6.1. Product Introduction

### 6.1.1. What is Container Service?

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

Container Service simplifies the deployment and scale-out operations of Kubernetes clusters and integrates Alibaba Cloud capabilities of virtualization, storage, networking, and security. Based on these capabilities, Container Service provides an ideal runtime environment for Kubernetes-based containerized applications. Alibaba Cloud is a Kubernetes Certified Service Provider (KCSP). As one of the first services to participate in the Certified Kubernetes Conformance Program, Container Service provides you with professional support and services.

#### Benefits

Container Service provides lifecycle management for enterprise-level containerized applications and enables automated, intelligent, simplified, and efficient application management and O&M. This allows you to run containerized applications in the cloud in a convenient and efficient manner.

#### Scenarios

- DevOps and continuous delivery

  Container Service and Jenkins can complete the DevOps process from code submission to application deployment with no need for manual operations. Only code that has been automatically tested can be delivered for deployment. This allows you to get rid of the traditional delivery modes that have complicated deployment and slow application iteration.

- Microservices Architecture

  Container Service can divide applications in production environments into microservices that are hosted in the repositories of Container Registry. These microservices can be separately deployed to achieve agile development and fast iteration.

- Hybrid cloud architecture

  You can manage both cloud and on-premises resources in the Container Service console. Container Service does not rely on specific infrastructure. Therefore, you can use the same images and orchestration templates to deploy applications both in the cloud and on the premises.

- Auto scaling architecture

Container Service can automatically scale your workloads based on the traffic load without the need for manual operations. This handles traffic spikes in time and prevents system failures caused by heavy traffic loads. Container Service provides a fully automated and fast auto scaling solution to reduce resource costs.

### Benefits

Container Service allows you to manage various types of clusters and automatically scale resources based on the requirements of your workloads. This provides an all-in-one solution to the management of IaaS, resources, and containers. Container Service can meet enterprise-level requirements for security and stability and provides 24/7 technical support.

# 6.1.2. 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.

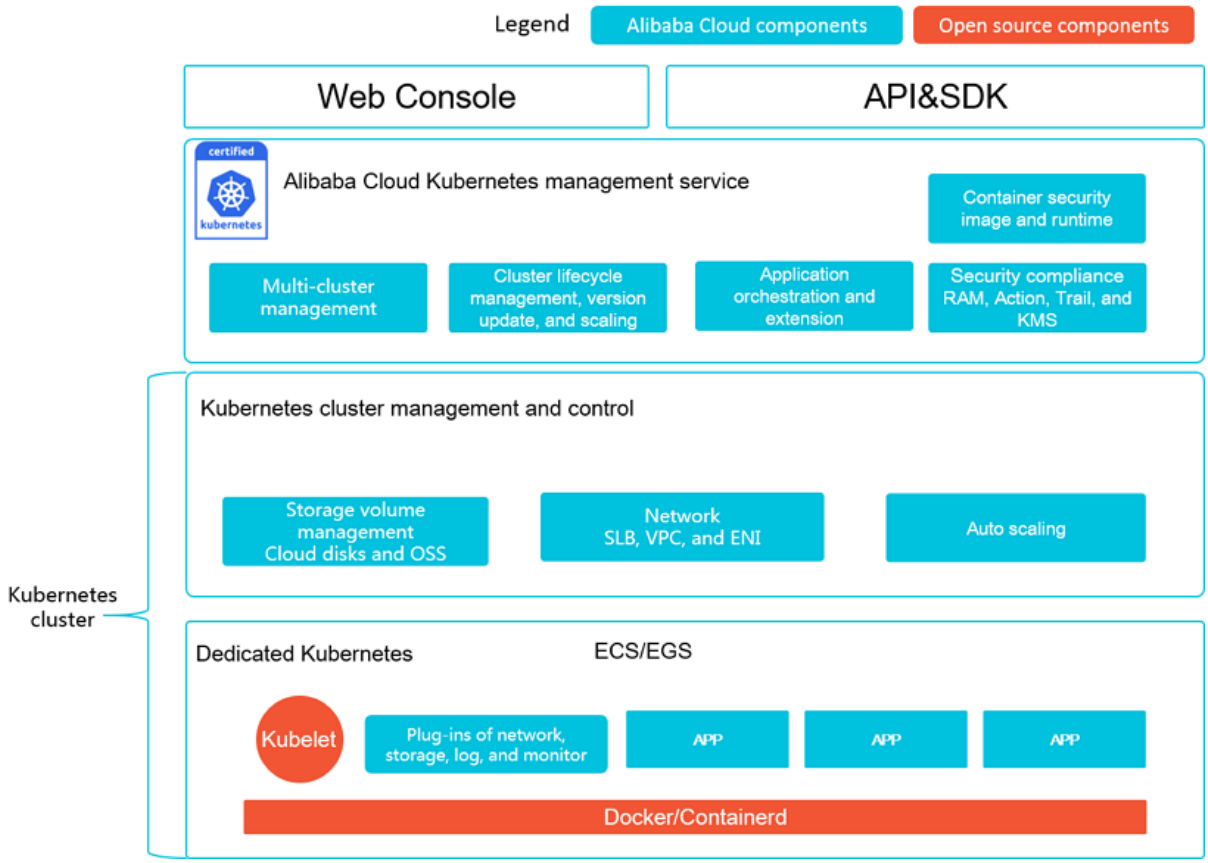
  Kubernetes clusters enable you to quickly scale up or down applications to handle traffic fluctuations in a timely manner.

**Features**

| Feature | Description |
| --- | --- |
| Network | Supports continuous network integration to optimize network performance. |
| Load balancing | Allows you to create public and internal SLB instances.<br>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. |

| Feature | Description |
| --- | --- |
| Storage | Supports Apsara Stack cloud disks, Network Attached Storage (NAS), and Block Storage, and provides FlexVolume drivers.<br>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.<br>• Supports automatic scaling. |
| Image repository | • Provides high availability and high concurrency.<br>• Supports accelerated image retrieval.<br>• Supports peer-to-peer image distribution.<br>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.<br>• All Linux and Kubernetes versions must pass rigorous testing before they are available to the public.<br>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. |
| Technical support | • Allows you to quickly upgrade Kubernetes clusters to the latest version.<br>• Provides professional technical support services to help you solve the issues that may occur when you use containers. |

# 6.1.3. 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. |
| 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), 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. |

# 6.1.4. Features

## Features of Container Service

**Cluster management**

- Allows you to create a dedicated Kubernetes cluster that contains GPU servers within 10 minutes in the Container Service console.
- Provides OS images that are optimized for containerized applications and supports Kubernetes versions and Docker versions with high stability and enhanced security.
- Supports multi-cluster management, cluster updates, and cluster scaling.

**Provides all-in-one container lifecycle management**

- **Network**

  Provides high-performance virtual private clouds (VPCs) and elastic network interfaces (ENIs) that are optimized for Alibaba Cloud, boasting 20% increased performance compared with regular network solutions.

  Supports access control and traffic throttling for containers.

- **Storage**

  Supports Alibaba Cloud disks and OSS buckets, and provides the standard FlexVolume driver.

  Supports dynamic volume creation and migration.

- **Logs**

  Provides high-performance log collection based on Log Service.

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

- **Monitoring**

  Supports container-level and VM-level monitoring. You can also integrate Container Service with third-party open source monitoring solutions.

- **Permissions**

  Supports cluster-level Resource Access Management (RAM) authorizations.

  Supports application-level permission configuration and management.

- **Application management**

  Supports canary release and blue-green releases.

  Supports application monitoring and scaling.

**High-availability scheduling policies that integrate upstream and downstream delivery processes**

- Supports service-level affinity policies and scale-out.
- Provides cross-zone high availability and disaster recovery.
- Provides API operations for cluster and application management to easily implement continuous integration and integrate with private deployment systems.

# 6.1.5. 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

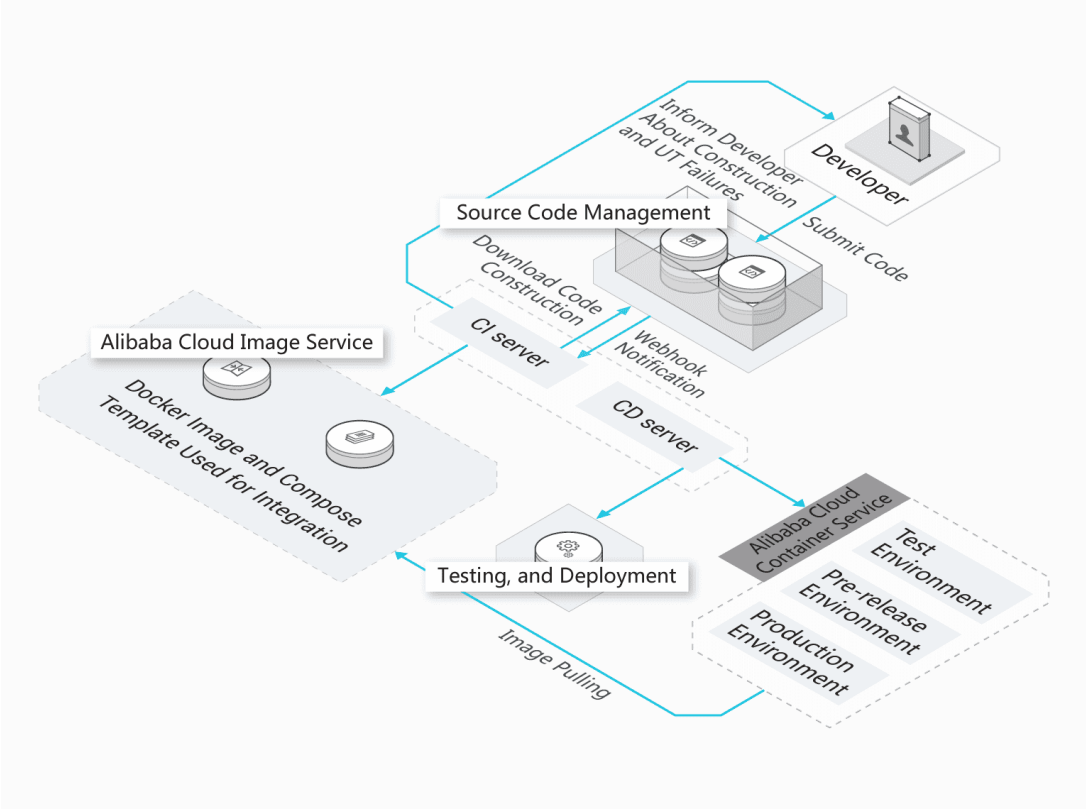
  Allows you to deliver code and runtime environments based on the same architecture.

- 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

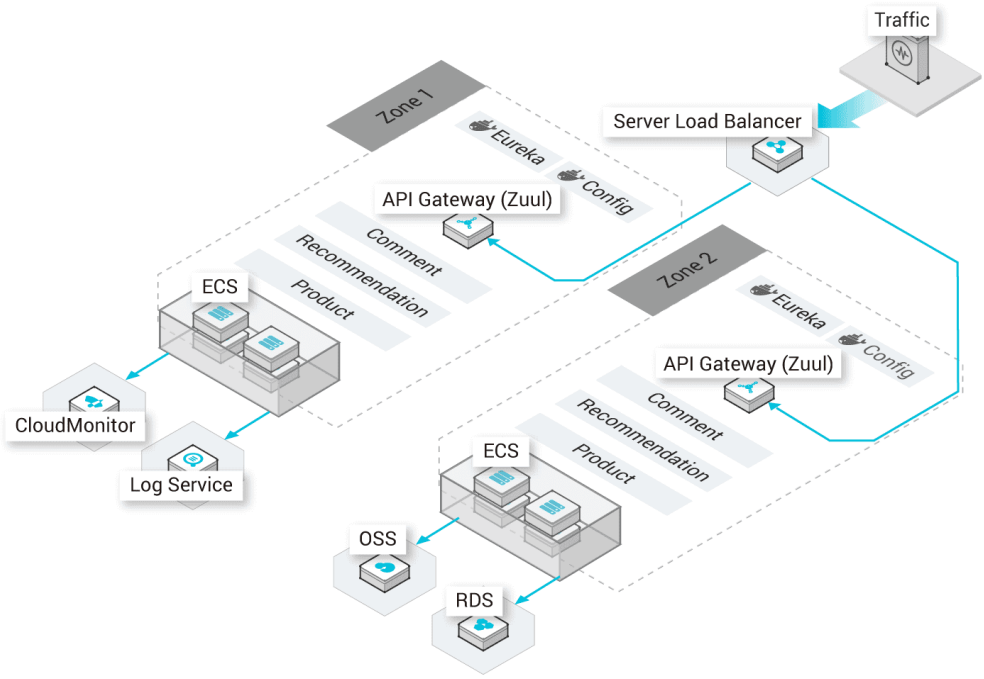
  Supports different levels of affinity scheduling policies, and cross-zone high availability and disaster recovery.

- 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

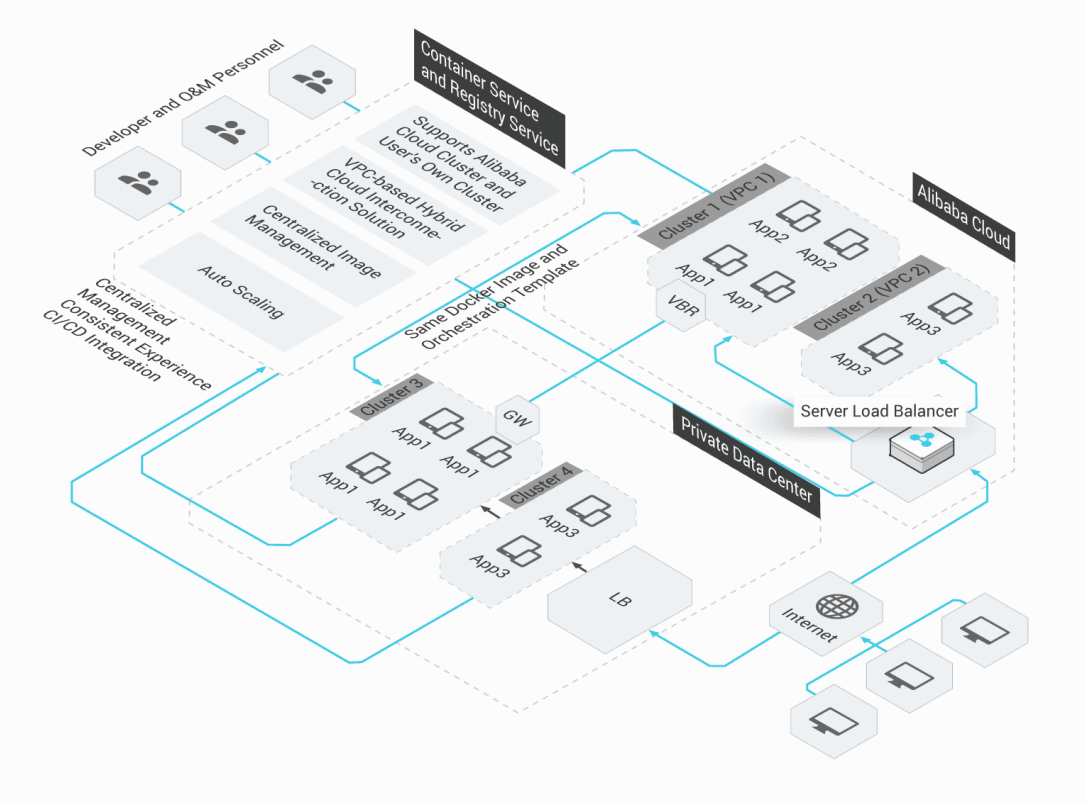
  Business systems can be deployed on premises for service provisioning and in the cloud for disaster recovery.

- 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

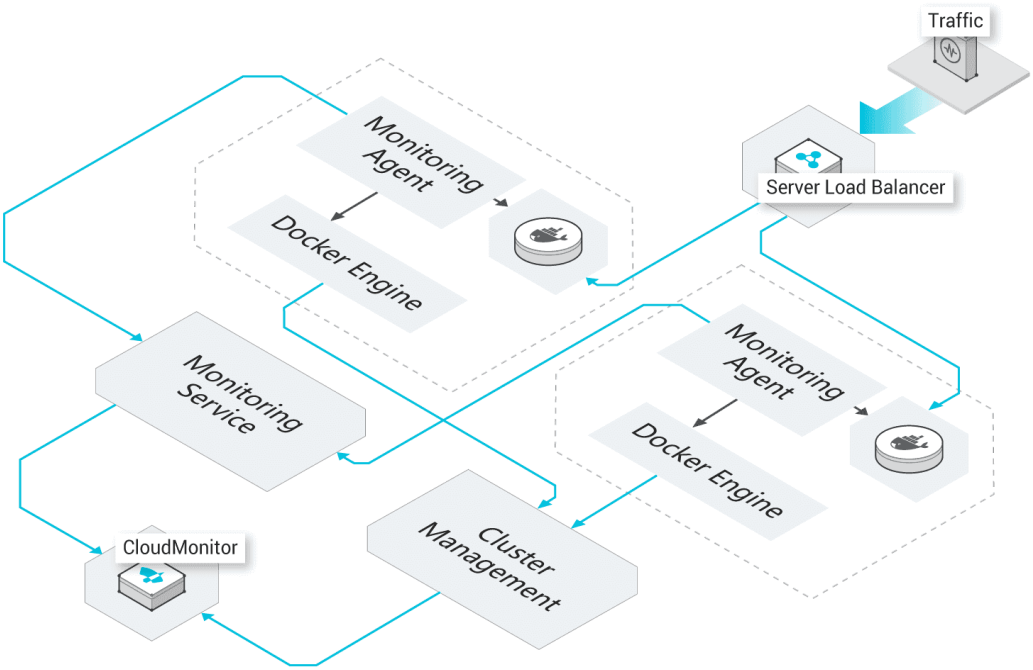
  The scaling process is fully automated without human interference.

- Low cost

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

### Related products and services

ECS + CloudMonitor




# 6.1.6. 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

# 7.Resource Orchestration Service (ROS)

## 7.1. Product Introduction

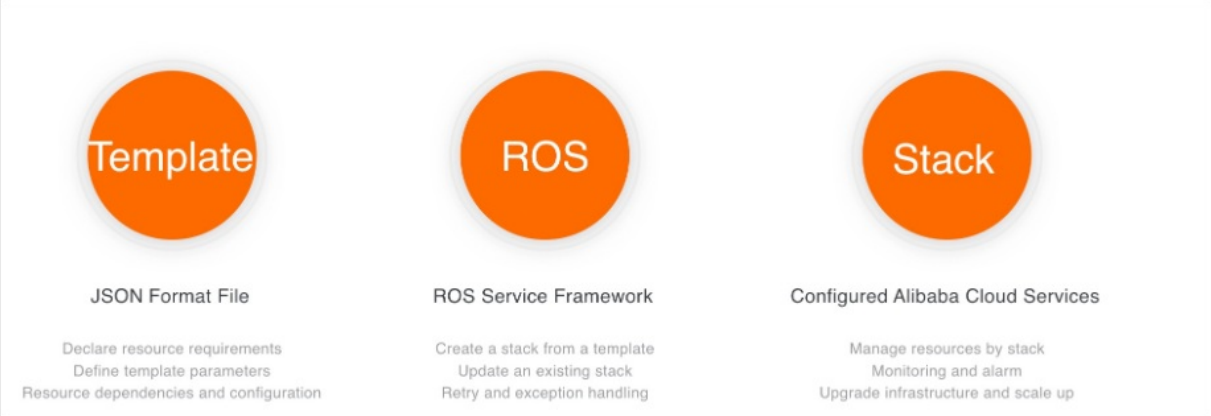
### 7.1.1. What is ROS?

Resource Orchestration Service (ROS) is a service provided by Apsara Stack to simplify the management of cloud computing resources. You can author a stack template based on the template specifications defined in ROS. In the 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 the template. This helps achieve automatic deployment and O&M.

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

An ROS template is a standardized method 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. This way, Apsara Stack resources can be integrated with your software systems for centralized delivery.

ROS manages a collection of cloud resources in a centralized manner by using a single unit named a stack. A stack is a collection of Apsara Stack resources. You can create, delete, and recreate cloud resources by stack.




### 7.1.2. Benefits

You can use Resource Orchestration Service (ROS) to model and configure your Apsara Stack resources.

After you create a template that defines your required resources such as Elastic Compute Service (ECS) and ApsaraDB 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) service provided by Alibaba Cloud to quickly implement IaC as a key component of DevOps.

### Fully managed automation service

ROS is a fully managed service. You do not need to purchase the resources that are used to maintain your templates. You need only to focus on maintaining the resources of your business and the template specifications. If 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 manage stacks and use source code versioning software such as Git and SVN to centrally manage your 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 between environments may lead to high management costs, prolonged troubleshooting time, and interruptions of the normal operation of your business. By using ROS for repeated deployment, you can standardize deployment environments, minimize differences between 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 peer 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 by means of APIs. You can use ActionTrail to review all O&M operations, including operations on ROS.

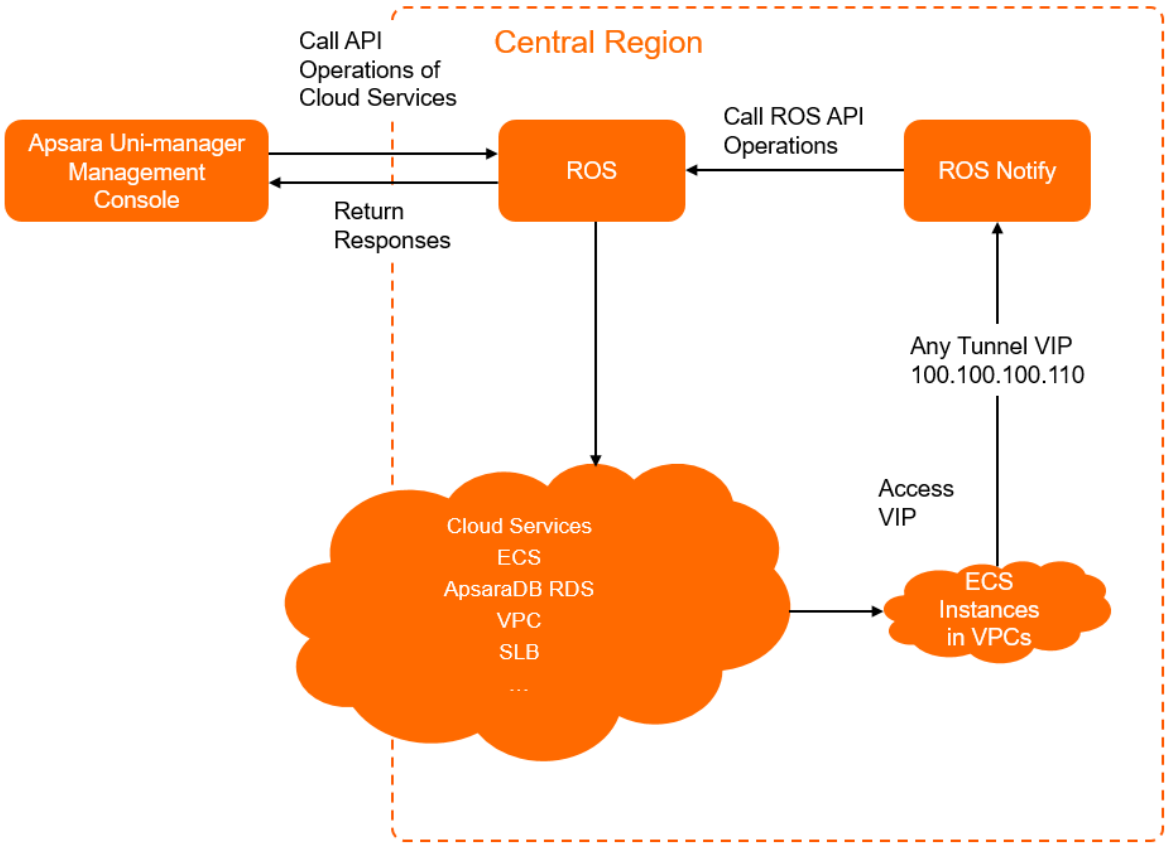
## 7.1.3. Architecture

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.

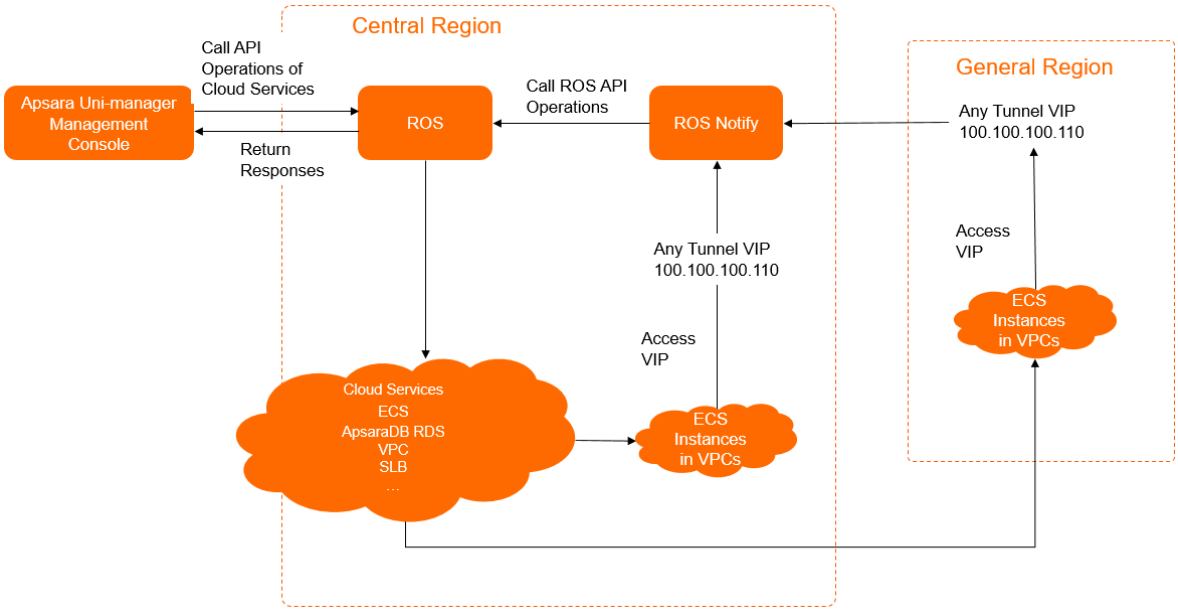
ROS supports the following cloud services: ECS, ApsaraDB RDS, ApsaraDB for MongoDB, ApsaraDB for Redis, ApsaraDB for Memcache, Server Load Balancer (SLB), Object Storage Service (OSS), Virtual Private Cloud (VPC), Elastic IP Address (EIP), Auto Scaling (ESS), Log Service, and Resource Access Management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

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

- Single-region deployment




- **Multi-region deployment**




# 7.1.4. 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 O&M possible. You can use ROS to build your own infrastructure on the cloud and implement the infrastructure as code (IaC). You do not need to call cloud service API operations to implement your infrastructure. ROS helps you process business resources in a more efficient way.

## Manage stacks

A stack is a collection of Apsara Stack resources that you can manage as a single unit. You can create, update, recreate, and delete a stack.

- Create a stack

  You can create a template that defines a set of resources such as Elastic Compute Service (ECS) and ApsaraDB RDS instances and the dependencies between the resources. Then, you can create a stack based on the template to manage your resources.

- Update a stack

  You can update a stack if you want to modify only the template that is used to create the stack or the parameter settings of the stack. The update operation does not change the organization, resource set, or region ID of your stack.

- Recreate a stack

  You can recreate a stack if you want to change your template, stack configurations, and the organization, resource set, and region ID of the stack.

- Delete a stack

  You can delete stacks that you no longer need. When you delete a stack, you can choose to retain or release resources in the stack based on your business requirements.

## Manage templates

A template is a UTF-8 encoded JSON file that is used to create stacks. Templates serve as the blueprint for infrastructure and architecture. You can define Apsara Stack resources, their configurations, and dependencies between the resources in a template.

- Create a template

  You can create a template in the ROS console. Then, you can use the template to create a stack.

- Edit a template

  You can edit the name, description, and content of a template based on your business requirements.

- Delete a template

  You can delete templates that you no longer need.

# 7.1.5. 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.

## 7.1.6. 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.

## 7.1.7. Terms

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

| Term | Description |
| --- | --- |
| template | A template is a UTF-8 encoded JSON file that is used to create stacks. Templates serve as the blueprint for infrastructure and architecture. Templates define the configurations and dependencies of Apsara Stack resources. |
| stack | A stack is a collection of Apsara Stack 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 Apsara Stack resources. |

# 8.Object Storage Service (OSS)

## 8.1. Product Introduction

### 8.1.1. What is OSS?

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

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 access control list (ACL) of the bucket or the object.
- Perform basic and advanced operations in the OSS console.
- Perform basic and advanced operations by using OSS SDKs or calling RESTful API operations in your application.

### 8.1.2. 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 self-managed server storage to help you better understand the benefits of OSS.

#### Advantages of OSS over self-managed server storage

| Item | OSS | Self-managed server storage |
| --- | --- | --- |
| Reliability | • Provides automatic backup for redundancy.<br>• Tolerates faults at the hard disk, node, rack, and cluster levels. Read and write operations are not interrupted in the event of failures of up to two nodes. This ensures business continuity. | • Is prone to errors due to low hardware reliability. If a disk has a bad sector, data may be irreversibly lost.<br>• Requires manual restoration of data, which can be a complex, time-consuming, and labor-intensive process. |

| Item | OSS | Self-managed server storage |
| --- | --- | --- |
| Security | • Provides multi-level security protection for enterprises.<br>• Provides resource isolation mechanisms for multiple tenants and supports zone-disaster recovery.<br>• Provides various authentication and authorization mechanisms. It also provides features such as allowlists, hotlink protection, Resource Access Management (RAM), and Security Token Service (STS) for temporary access. | • Requires additional scrubbing devices and blackhole policy-related services.<br>• Requires a separate security mechanism. |
| Data processing | Provides Image Processing (IMG). | Requires separate purchase and deployment of data processing capabilities. |

## More benefits of OSS

- Ease of use
  - OSS provides standard RESTful API operations, some of which are compatible with Amazon S3 API operations, a wide range of SDKs, client tools, and the OSS console. You can use any one of these options to upload, download, query, and manage large amounts of data used in your apps and websites in the same way you would with regular file systems.
  - OSS supports streaming writes and reads. It is suitable for business scenarios that require simultaneous write and read of large files such as videos.
  - OSS supports lifecycle management. You can configure lifecycle rules to delete expired objects in batches.
  - OSS provides plenty of storage space that is also scalable. You can add nodes to increase your storage space. A single bucket can contain trillions of objects.
- Powerful and flexible security mechanisms

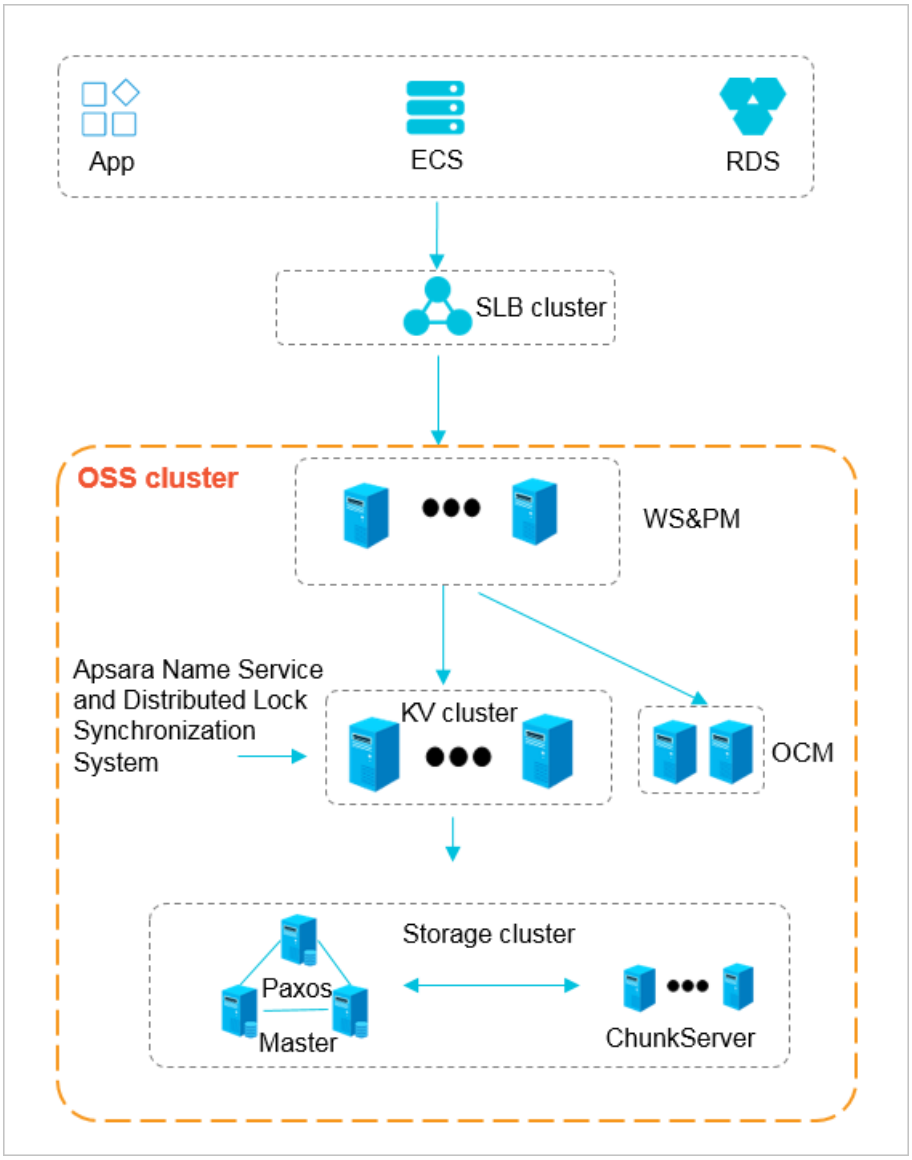
  OSS provides STS and URL-based authentication and authorization mechanisms, allowlists, hotlink protection, and RAM.
- Rich image processing features

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

# 8.1.3. 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.

# 8.1.4. 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.

# 8.1.5. Features

## 8.1.5.1. Manage buckets

## 8.1.5.1.1. Create a bucket

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

### Naming conventions

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

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

### Examples

The following examples of bucket names are valid:

- examplebucket1

- test-bucket-2021
- aliyun-oss-bucket

The following examples show invalid bucket names and the reasons why the names are invalid:

- Examplebucket1 (Uppercase letters are included.)
- test_bucket_2021 (Underscores (_) are included.)
- aliyun-oss-bucket- (The name ends with a hyphen (-).)

## 8.1.5.1.2. ACL

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

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

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

## 8.1.5.1.3. Static website hosting

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

## Usage notes

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

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

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

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

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

## Configurations

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

```
Bucket
  ├─── index.html
  ├─── error.html
  ├─── example.txt
  └─── subdir/
     └─── index.html
```

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

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

# 8.1.5.1.4. Logging

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

## Naming conventions for log objects

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

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

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

## Usage notes

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

# 8.1.5.1.5. Lifecycle rules

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

## Scenarios

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

## Usage notes

- Number of lifecycle rules

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

- Effective time

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

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

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

## Elements of a lifecycle rule

A lifecycle rule consists of the following elements:

- Policy: the policy used to match objects and parts.
  - Match by prefix: Objects and parts are matched by prefix. You can create multiple rules to match objects with different object name prefixes. Each prefix must be unique.
  - Match by tag: Objects are matched by tag key and tag value. You can specify multiple tags in a single lifecycle rule. The lifecycle rule applies to all objects that have the specified tags. Lifecycle rules cannot match parts by tag.
  - Match by prefix and tag: Objects are matched by specified prefixes and tags.
  - Match by bucket: The rule matches all objects and parts stored in the bucket. After you configure a lifecycle rule for a bucket to match all objects and parts in the bucket, other lifecycle rules cannot be configured for the bucket.
- Object lifecycle policy: specifies the validity period or the expiration date of objects and the operation to perform on expired objects.

  - Validity period: A validity period is specified for objects in buckets for which versioning is disabled and the current versions of objects in buckets for which versioning is enabled. In addition, the operation to perform on these objects after they expire is specified. Objects that match the lifecycle rule are retained for the specified validity period after the objects are last modified. The specified operation is performed on these objects after they expire.
  - Expiration date: An expiration date is specified for objects in buckets for which versioning is disabled and the current versions of objects in buckets for which versioning is enabled. In addition, the operation to perform on these objects after they expire is specified. All objects that are last modified before this date expire, and the specified operation is performed on these objects.
  - Validity period of the previous versions of objects: A validity period is specified for the previous versions of objects. In addition, the operation to perform on these previous versions is specified. Objects that match the lifecycle rule are retained for the specified validity period after the object versions become the non-curent versions. The specified operation is performed on these objects after they expire.
- Part lifecycle policy: the policy used to specify the validity period or expiration date for parts and the operation to perform on these expired parts.
  - Validity period: A validity period is specified for parts. Parts that match the lifecycle rule are retained within the validity period and are deleted after they expire.
  - Expiration date: An expiration date is specified for parts. Parts that are last modified before this date expire and are deleted.

# 8.1.5.2. Manage objects

# 8.1.5.2.1. Upload objects

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

OSS provides the following upload methods:

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

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

# 8.1.5.2.2. ACL

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

You can configure ACL for an object when or after you upload the object. By default, if you do not specify ACL for an object, the ACL of the object is **Inherited from Bucket**.

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

# 8.1.5.2.3. Download objects

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

OSS provides the following object download methods:

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

# 8.1.5.2.4. Search for objects

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

## Usage notes

- Search rules

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

- Search results

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

### Examples

- Search for specific objects or directories within the root directory of the bucket

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

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

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

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

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

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

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

# 8.1.5.2.5. Manage objects by using directories

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

### Structure

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

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

In the preceding structure:

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

## Access control based on directories

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

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

## Implementation methods

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

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

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

# 8.1.5.2.6. Object tagging

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

## Usage notes

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

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

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

## Configure lifecycle rules for objects with the same tags

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

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

Example:

```
<LifecycleConfiguration>
  <Rule>
    <ID>r1</ID>
    <Prefix>rule1</Prefix>
    <Tag><Key>xx</Key><Value>1</Value></Tag>
    <Tag><Key>yy</Key><Value>2</Value></Tag>
    <Status>Enabled</Status>
    <Expiration>
      <Days>30</Days>
    </Expiration>
  </Rule>
  <Rule>
    <ID>r2</ID>
    <Prefix>rule2</Prefix>
    <Tag><Key>xx</Key><Value>1</Value></Tag>
    <Status>Enabled</Status>
    <Transition>
      <Days>60</Days>
      <StorageClass>Archive</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

In the preceding rules:

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

## Use RAM policies to manage permissions on objects with specified tags

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

- Authorize RAM users to manage object tags

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

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

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

- Authorize RAM users to manage objects that have specific tags

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

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

# 8.1.5.3. Data security

# 8.1.5.3.1. Erasure coding

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

## EC

EC involves the following two concepts:

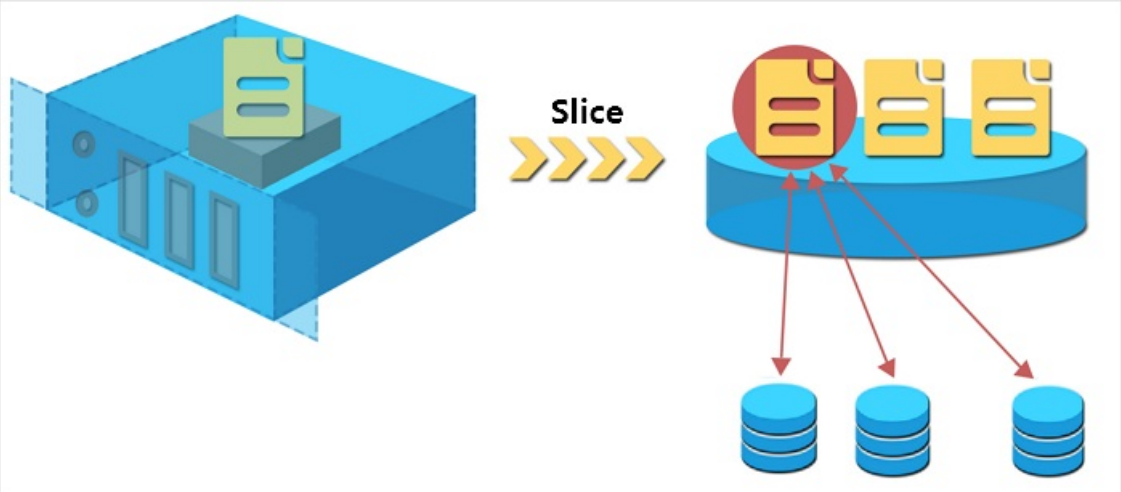
- Data fragments (m): Data is divided into m data fragments.
- Parity fragments (n): n parity fragments are computed based on the m data fragments.

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

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

## Triplicate

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




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

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

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

### Comparison between EC and triplicate storage

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

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

## 8.1.5.3.2. Resource isolation

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

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

## 8.1.5.3.3. Disaster recovery

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

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

| Type | Description |
| --- | --- |
| Zone-disaster recovery | Zone-disaster recovery allows you to store multiple replicas of your data in multiple zones within the same region. This feature protects your data from being lost and helps you recover your business when a single zone fails. |
| CRR | Cross-region replication (CRR) enables the automatic and asynchronous (near real-time) replication of objects across OSS buckets in different regions. Operations such as the creation, overwriting, and deletion of objects can be synchronized from a source bucket to a destination bucket. |

| Type | Description |
| --- | --- |
| Cross-cloud replication | You can use cross-cloud replication to replicate data from one cloud to another cloud. This way, you can back up data across clouds. If a cloud fails, you can switch over your business to another cloud to ensure business continuity. |
| Three data centers across two regions | If your business has high requirements on data backup, you can use zone-disaster recovery and cross-region replication to build a disaster recovery solution based on three data centers across two regions. |

# 8.1.5.3.4. Access permissions and account authorization

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

**OSS provides the following access permission policies.**

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

# 8.1.5.3.5. Server-side encryption

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

## Encryption methods

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

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

## Server-side encryption by using OSS-managed keys

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

Use the following configuration methods:

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

  Set the default encryption method to SSE-OSS and specify the encryption algorithm as AES-256 . This way, all objects uploaded to this bucket are encrypted.

- Configure an encryption method for a specific object

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

# 8.1.5.3.6. Client-side encryption

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

## Disclaimer

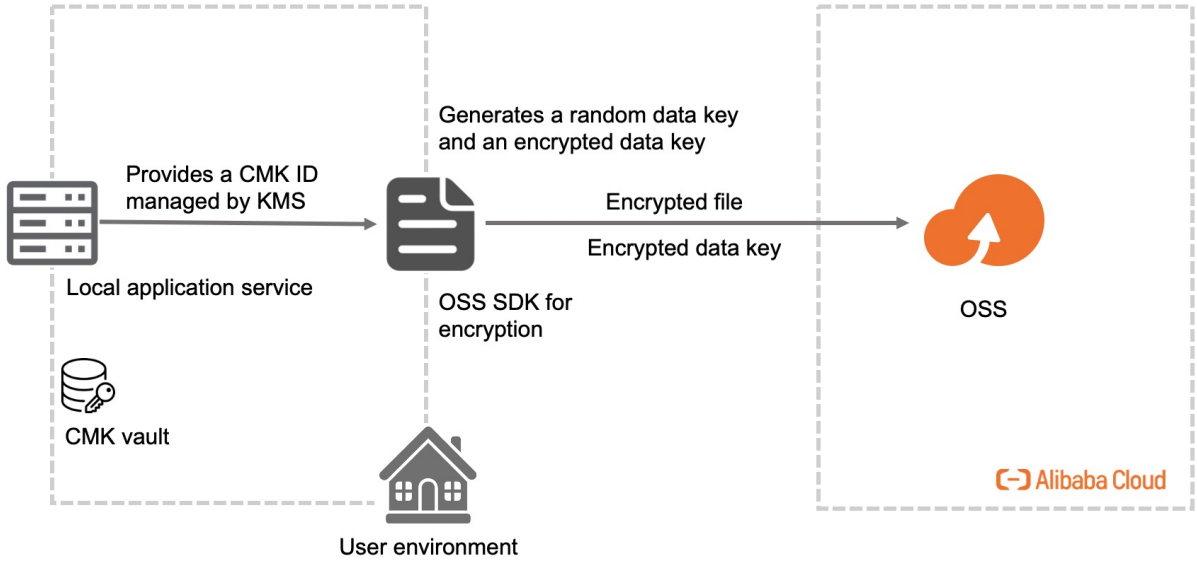
- When you use client-side encryption, you must ensure the integrity and validity of the customer master key (CMK). If the CMK is incorrectly used or lost due to improper key management, you are responsible for all losses and consequences caused by decryption failures.
- When you copy or migrate encrypted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

## Encryption

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

## Use customer-managed CMK

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




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

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

### Usage notes

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

# 8.1.5.4. Data processing

# 8.1.5.4.1. IMG

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

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

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

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

| IMG operation | Parameter | Description |
| --- | --- | --- |
| Blur | blur | Blurs images. |
| Adjust brightness | bright | Adjusts the brightness of images. |
| Sharpen | sharpen | Sharpens images. |
| Adjust contrast | contrast | Adjusts the contrast of images. |
| Gradual display | interlace | Configures gradual display for the JPG images. |
| Adjust image quality | quality | Adjusts the quality of JPG and WebP images. |
| Convert format | format | Converts image formats. |
| Add watermarks | watermark | Adds image or text watermarks to images. |
| Query average tone | average-hue | Queries the average tone of images. |
| Query image information | info | Queries image information, including basic information and Exchangeable Image File Format (EXIF) information. |

# 8.1.5.4.2. Video snapshots

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

## Usage notes

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

## Parameters

Operation type: `video`

Operation name: snapshot

| Parameter | Description | Valid value |
| --- | --- | --- |
| t | Specifies the time when the image is to be captured. | [0, video duration]<br>Unit: ms |

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

### Examples

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

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

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

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

# 8.1.6. 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. Various devices, websites, and mobile applications can directly write data to and read data from OSS. You can write data to OSS by uploading files or using streams.

## Offline data storage

OSS is a cost-effective storage service that offers high data availability. You can use OSS to store enterprise data that needs to be archived offline for a long period of time.

## Cross-region disaster recovery

You can use cross-region replication (CRR) or cross-cloud replication to asynchronously replicate your data between two clusters or clouds in near real time. This way, you can build a storage architecture with three data centers in two regions and store your data in different regions for backup and disaster recovery. This ensures the continuity of your business even in case of severe disaster events.

# 8.1.7. Limits

This topic describes the limits and performance metrics of OSS.

| Item | Limit |
| --- | --- |
| Bucket | • You can create a maximum of 100 buckets.<br>• 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.<br>• If you upload an object that has the same name of an existing object in OSS, the new object will overwrite the existing object.<br>• OSS traffic is forwarded through SLB and has the following limits:<br>  ○ By default, a virtual IP address (VIP) is configured for SLB and the maximum throughput for OSS is 1.25 GB/s.<br>  ○ 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.<br>• 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. |

# 9.Cloud Defined Storage (CDS)

# 9.1. Product Introduction

## 9.1.1. What is CDS?

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

### Overview

You can deploy cloud services, such as Object Storage Service (OSS), Apsara File Storage NAS (NAS), Elastic Block Storage (EBS), and Log Service in CDS. Then, you can use CDS to store unstructured data such as files, images, and videos, and store, query, and analyze log data. This helps you meet the requirements in different industries. CDS is applicable to big data scenarios such as mobile applications and large websites where you need to access unstructured data and process massive logs. CDS can provide one-stop storage solutions for enterprises in different industries in a cost-efficient, secure, and reliable manner.

### Benefits

- Flexible deployment

  CDS defines storage based on cloud services, and allows you to deploy cloud services together or separately. In addition, CDS supports dynamic and flexible resource scaling.

- High performance
  - CDS can integrate resources such as CPU resources and hard disk resources on all storage nodes, and distribute data storage and process tasks in real time in a dynamic and balanced manner. This implements concurrent data processing, prevents issues caused by single points of failure (SPOFs), and improves the processing capabilities of clusters.
  - The performance of CDS storage clusters can be easily enhanced and can meet the growing storage requirements of applications.

- High reliability

  The storage clusters in CDS use the erasure coding mechanism to store data as data blocks on different servers in different racks. If a data block error occurs, you can restore the corresponding data block with ease.

- High availability
  - The storage clusters in CDS use fully redundant architectures. This prevents SPOFs. In addition, CDS provides automatic failure detection and data migration features to shield applications from server- and network-related hardware faults. This ensures high availability of applications.
  - The erasure coding mechanism used by storage clusters ensures proper data redundancy and enhances space utilization.

- High scalability

  CDS improves the service capability of a storage cluster by using various methods, such as expanding the cluster, adding and upgrading server hardware in the cluster, and add storage nodes to the cluster. The storage clusters in CDS support smooth online upgrades and hot upgrades, and allow you to dynamically add or remove storage nodes. In addition, the storage clusters support automatic data migration and do not require shutdown maintenance.

- Access control

  CDS provides multiple permission management mechanisms and authenticates each request from applications to prevent unauthorized access. This ensures data security.

- Easy management
  - CDS frees you from complex O&M tasks, such as the management of data partitions, software and hardware upgrades, configuration updates, and cluster scale-outs.
  - CDS allows you to store audit logs to Log Service and download logs from Log Service. This facilitates the long-term storage and management of audit logs.
  - The unified O&M platform CDS Ops can perform daily O&M operations on the storage clusters and cloud services deployed in these clusters, such as OSS, NAS, EBS, and Log Service.

## Features

You can deploy OSS and Log Service in storage clusters of CDS. The supported features of a storage cluster vary based on the service that you deployed in the cluster. The following tables describe the supported features by service.

- OSS

| Feature | Description |
| --- | --- |
| Bucket and object management | Before you can upload objects to OSS, you must create a bucket to store objects. After you create a bucket, you can manage the objects in the bucket based on bucket configurations such as hotlink protection and lifecycle. |
| Object upload and download | You can upload all types of objects to a bucket. After objects are uploaded to a bucket, you can share and download the uploaded objects based on the URLs of the objects. You can obtain the URLs of the uploaded objects one by one or at a time. |
| Access control | OSS provides access control lists (ACLs) for access control. An ACL is an access policy that authorizes access to buckets and objects. You can configure an ACL when you create a bucket or upload an object, or modify the ACL at any time after you create a bucket or upload an object. |
| Static website hosting | Static websites are websites in which all web pages consist only of static content, including scripts such as JavaScript code that can be run on the client. You can use the static website hosting feature to host your static website on an OSS bucket and use the endpoint of the bucket to access the website. |

| Feature | Description |
| --- | --- |
| Hotlink protection | The hotlink protection feature allows you to configure a Referer allowlist for a bucket to prevent your resources in the bucket from unauthorized access. This way, only requests from the domain names that are included in the Referer allowlist can access the data in the bucket. |
| Log management | When you access OSS, large numbers of access logs are generated. You can enable and configure logging for a bucket so that OSS generates logs every hour based on the defined naming rule and stores the logs as objects in the specified bucket.<br>The real-time log query feature integrates OSS with Log Service and allows you to directly query OSS access logs. You can use this feature to audit access to OSS, track exception events, and troubleshoot problems. This helps you improve work efficiency and make informed decisions. |
| Cross-origin resource sharing (CORS) | CORS is a standard cross-origin solution that is provided by HTML5 to allow web application servers to manage cross-origin access. This secures data transmission. |
| Lifecycle management | You can configure lifecycle rules to periodically delete expired files. This saves storage costs. |
| Image management | You can use the image style feature to add multiple image processing parameters to an image style to perform complex operations on images.<br>To prevent images that allow anonymous access in an OSS bucket from unauthorized use, you can enable source image protection for the bucket. After you enable source image protection for the bucket, anonymous requesters can access the images in the bucket only by adding style parameters in the requests or by using signed URLs. |

- Log Service

| Feature | Description |
| --- | --- |
| Real-time log collection and consumption (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. |
| Real-time log query and analysis (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. |

| Feature | Description |
| --- | --- |
| Alert management | You can create an alert rule for the query and analysis results. After you create an alert rule, Log Service checks the related query and analysis results on a regular basis. If a query and analysis result meets the trigger condition that you specified in the alert rule, Log Service sends an alert notification. This way, the service status is monitored in real time. |

- NAS

| Feature | Description |
| --- | --- |
| File system management | You can use the NAS console to manage the file systems within your Alibaba Cloud account. You can create a file system, delete a file system, and query the details of a file system. |
| Mount target management | A mount target is an access point of a NAS file system in the classic network or a virtual private cloud (VPC). Each mount target is displayed as a domain name. You can view the mount target of a file system and modify the status and permission group of the mount target in the NAS console. |
| Permission control and ACL-based isolation | You can configure directory-level ACLs for a NAS file system. You can configure ACLs for files or directories to control access by directory in a fine-grained manner. An administrator of a NAS file system can grant users and groups different access permissions on directories and files for access control. |

### How to use CDS

- Use the Apsara Uni-manager Management Console

  You can log on to the Apsara Uni-manager Management Console to use the features of services deployed in CDS. This facilitates service management.

- Use service SDKs

  You can use the SDKs of different services to flexibly use the corresponding service features in CDS. SDKs are implemented based on service APIs and provide capabilities equivalent to service APIs.

# 9.1.2. OSS

## 9.1.2.1. Product Introduction

### 9.1.2.1.1. What is OSS?

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

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 access control list (ACL) of the bucket or the object.
- Perform basic and advanced operations in the OSS console.
- Perform basic and advanced operations by using OSS SDKs or calling RESTful API operations in your application.

# 9.1.2.1.2. 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 self-managed server storage to help you better understand the benefits of OSS.

## Advantages of OSS over self-managed server storage

| Item | OSS | Self-managed server storage |
| --- | --- | --- |
| Reliability | • Provides automatic backup for redundancy.<br>• Tolerates faults at the hard disk, node, rack, and cluster levels. Read and write operations are not interrupted in the event of failures of up to two nodes. This ensures business continuity. | • Is prone to errors due to low hardware reliability. If a disk has a bad sector, data may be irreversibly lost.<br>• Requires manual restoration of data, which can be a complex, time-consuming, and labor-intensive process. |
| Security | • Provides multi-level security protection for enterprises.<br>• Provides resource isolation mechanisms for multiple tenants and supports zone-disaster recovery.<br>• Provides various authentication and authorization mechanisms. It also provides features such as allowlists, hotlink protection, Resource Access Management (RAM), and Security Token Service (STS) for temporary access. | • Requires additional scrubbing devices and blackhole policy-related services.<br>• Requires a separate security mechanism. |

| Item | OSS | Self-managed server storage |
| --- | --- | --- |
| Data processing | Provides Image Processing (IMG). | Requires separate purchase and deployment of data processing capabilities. |

### More benefits of OSS

- Ease of use
  - OSS provides standard RESTful API operations, some of which are compatible with Amazon S3 API operations, a wide range of SDKs, client tools, and the OSS console. You can use any one of these options to upload, download, query, and manage large amounts of data used in your apps and websites in the same way you would with regular file systems.
  - OSS supports streaming writes and reads. It is suitable for business scenarios that require simultaneous write and read of large files such as videos.
  - OSS supports lifecycle management. You can configure lifecycle rules to delete expired objects in batches.
  - OSS provides plenty of storage space that is also scalable. You can add nodes to increase your storage space. A single bucket can contain trillions of objects.
- Powerful and flexible security mechanisms

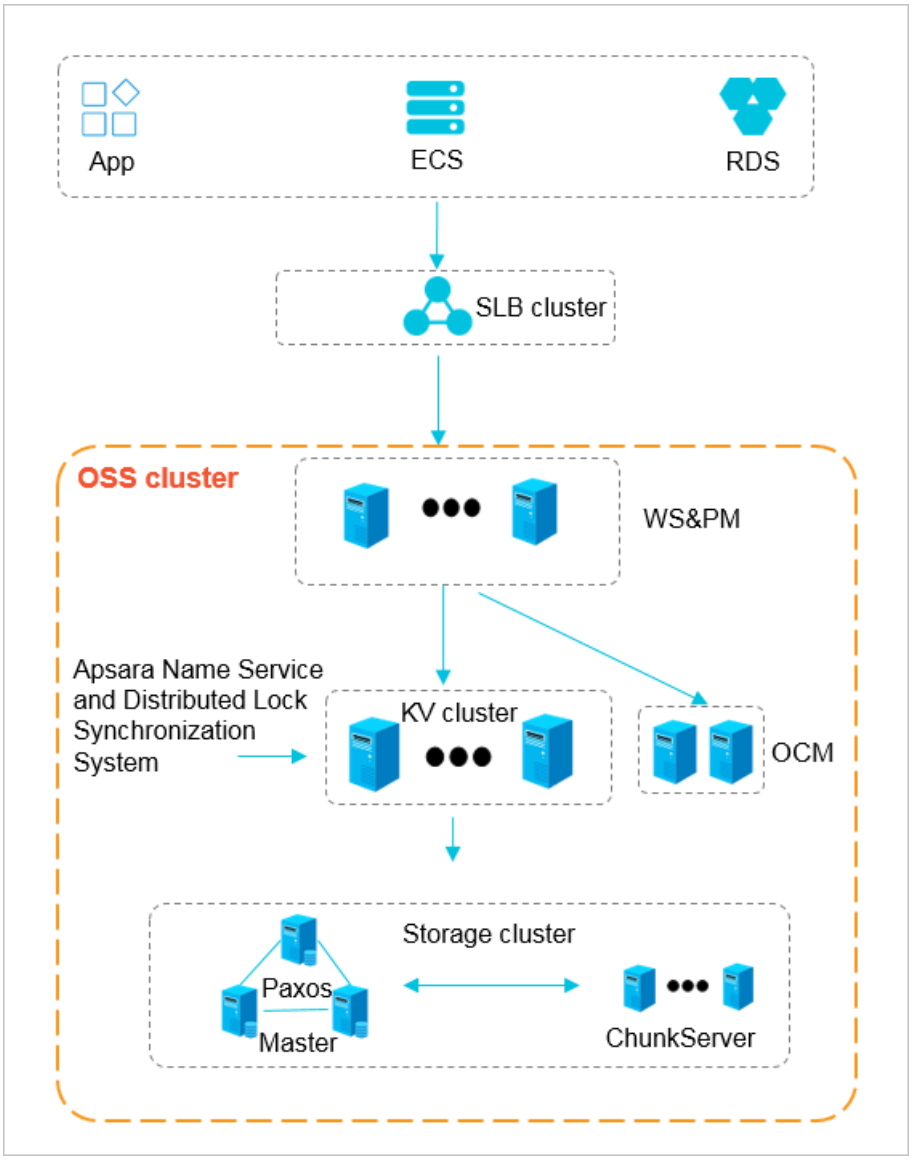
  OSS provides STS and URL-based authentication and authorization mechanisms, allowlists, hotlink protection, and RAM.
- Rich image processing features

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

## 9.1.2.1.3. 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.

# 9.1.2.1.4. 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.

# 9.1.2.1.5. Features

# 9.1.2.1.5.1. Manage buckets

Create a bucket

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

## Naming conventions

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

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

## Examples

The following examples of bucket names are valid:

- examplebucket1

- test-bucket-2021
- aliyun-oss-bucket

The following examples show invalid bucket names and the reasons why the names are invalid:

- Examplebucket1 (Uppercase letters are included.)
- test_bucket_2021 (Underscores (_) are included.)
- aliyun-oss-bucket- (The name ends with a hyphen (-).)

ACL

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

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

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

Static website hosting

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

## Usage notes

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

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

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

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

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

## Configurations

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

```
Bucket
  ├─── index.html
  ├─── error.html
  ├─── example.txt
  └─── subdir/
     └─── index.html
```

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

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

Logging

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

## Naming conventions for log objects

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

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

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

## Usage notes

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

Lifecycle rules

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

## Scenarios

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

## Usage notes

- Number of lifecycle rules

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

- Effective time

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

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

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

## Elements of a lifecycle rule

A lifecycle rule consists of the following elements:

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

  - Expiration date: An expiration date is specified for objects in buckets for which versioning is disabled and the current versions of objects in buckets for which versioning is enabled. In addition, the operation to perform on these objects after they expire is specified. All objects that are last modified before this date expire, and the specified operation is performed on these objects.
  - Validity period of the previous versions of objects: A validity period is specified for the previous versions of objects. In addition, the operation to perform on these previous versions is specified. Objects that match the lifecycle rule are retained for the specified validity period after the object versions become the non-curent versions. The specified operation is performed on these objects after they expire.

- Part lifecycle policy: the policy used to specify the validity period or expiration date for parts and the operation to perform on these expired parts.
  - Validity period: A validity period is specified for parts. Parts that match the lifecycle rule are retained within the validity period and are deleted after they expire.
  - Expiration date: An expiration date is specified for parts. Parts that are last modified before this date expire and are deleted.

# 9.1.2.1.5.2. Manage objects

Upload objects

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

OSS provides the following upload methods:

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

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

ACL

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

You can configure ACL for an object when or after you upload the object. By default, if you do not specify ACL for an object, the ACL of the object is **Inherited from Bucket**.

- **Inherited from Bucket**: The ACL for the object is the same as that for the bucket.
- **Private**: Only the bucket owner or authorized users can read from and write to the objects in the bucket. Other users, including anonymous users, cannot access objects in the bucket.

- **Public Read**: Only the bucket owner or authorized users can read from and write to objects. Other users, including anonymous users, can only read from objects in the bucket.
- **Public Read/Write**: All users, including anonymous users, can perform read and write operations on objects in the bucket. The bucket owner are charged fees incurred by these operations. Therefore, we recommend that you use this ACL policy only when necessary.

Download objects

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

OSS provides the following object download methods:

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

Search for objects

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

## Usage notes

- Search rules

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

- Search results

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

## Examples

- Search for specific objects or directories within the root directory of the bucket

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

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

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

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

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

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

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

Manage objects by using directories

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

## Structure

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

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

In the preceding structure:

- The following three objects have the log prefix in their names: log/date1.txt, log/date2.txt, and log/date3.txt. In the OSS console, a directory named log is displayed. Three objects named date1.txt, date2.txt, and date3.txt are stored within the directory.

- The destfolder/2021/photo.jpg object has the destfolder prefix in its name. In the OSS console, a directory named destfolder is displayed, which contains a subdirectory named 2021. An object named photo.jpg is stored in the 2021 subdirectory.

## Access control based on directories

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

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

## Implementation methods

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

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

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

Object tagging

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

## Usage notes

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

- A maximum of 10 tags can be configured for each object. Tags associated with an object must have unique tag keys.
- A tag key can be up to 128 bytes in length. A tag value can be up to 256 bytes in length.
- Tag keys and tag values are case-sensitive.

- The key and the value of a tag can contain letters, digits, spaces, and the following special characters:

  + - = . _ : /

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

## Configure lifecycle rules for objects with the same tags

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

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

Example:

```
<LifecycleConfiguration>
 <Rule>
  <ID>r1</ID>
  <Prefix>rule1</Prefix>
  <Tag><Key>xx</Key><Value>1</Value></Tag>
  <Tag><Key>yy</Key><Value>2</Value></Tag>
  <Status>Enabled</Status>
  <Expiration>
    <Days>30</Days>
  </Expiration>
 </Rule>
 <Rule>
  <ID>r2</ID>
  <Prefix>rule2</Prefix>
  <Tag><Key>xx</Key><Value>1</Value></Tag>
  <Status>Enabled</Status>
  <Transition>
   <Days>60</Days>
   <StorageClass>Archive</StorageClass>
 </Transition>
</Rule>
</LifecycleConfiguration>
```

In the preceding rules:

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

## Use RAM policies to manage permissions on objects with specified tags

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

- Authorize RAM users to manage object tags

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

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

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

- Authorize RAM users to manage objects that have specific tags

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

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

# 9.1.2.1.5.3. Data security

Erasure coding

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

## EC

EC involves the following two concepts:

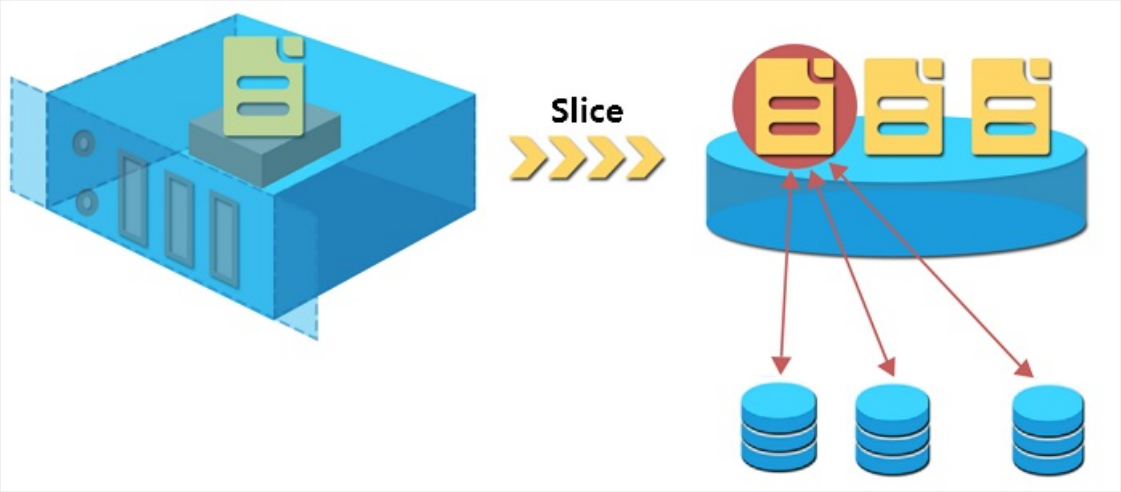
- Data fragments (m): Data is divided into m data fragments.
- Parity fragments (n): n parity fragments are computed based on the m data fragments.

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

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

## Triplicate

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




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

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

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

## Comparison between EC and triplicate storage

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

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

Resource isolation

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

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

Disaster recovery

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

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

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

Access permissions and account authorization

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

OSS provides the following access permission policies.

| Policy | Description |
| --- | --- |
| RAM Policy | Resource Access Management (RAM) is a service provided by Alibaba Cloud to manage access permissions on resources. RAM policies are configured based on users. You can manage users by configuring RAM policies. For users such as employees, systems, or applications, you can control which resources are accessible. For example, you can create a RAM policy to grant users only read permissions on a bucket. |

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

Server-side encryption

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

## Encryption methods

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

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

## Server-side encryption by using OSS-managed keys

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

Use the following configuration methods:

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

Set the default encryption method to SSE-OSS and specify the encryption algorithm as AES-256 . This way, all objects uploaded to this bucket are encrypted.

- Configure an encryption method for a specific object

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

Client-side encryption

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

## Disclaimer

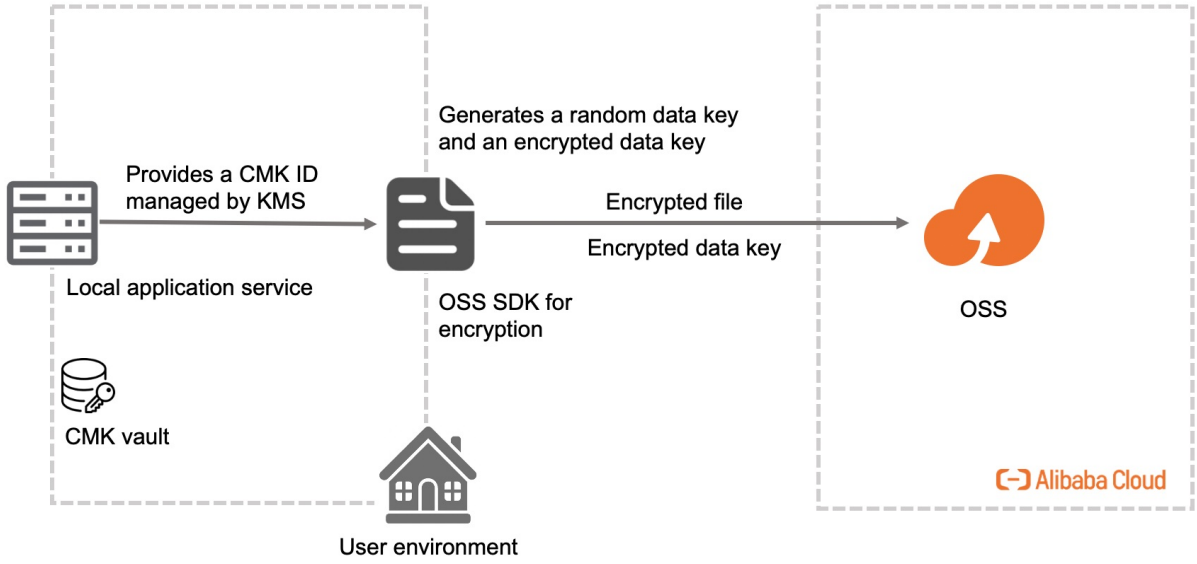
- When you use client-side encryption, you must ensure the integrity and validity of the customer master key (CMK). If the CMK is incorrectly used or lost due to improper key management, you are responsible for all losses and consequences caused by decryption failures.
- When you copy or migrate encrypted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

## Encryption

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

## Use customer-managed CMK

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




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

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

## Usage notes

- To perform client-side encryption on objects that are larger than 5 GB in size before you upload the objects, you must use multipart upload. When you use multipart upload to upload an

object, you must specify the total size of the object and the size of each part. The size of each part except for the last part must be the same and be a multiple of 16 bytes.

- After you upload objects encrypted on the client, object metadata related to client-side encryption is protected. In this case, CopyObject cannot be called to modify object metadata.

Versioning

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

## Versioning states

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

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

## Scenarios

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

- Recover deleted data

  You can configure versioning to recover deleted data.

- Recover overwritten data

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

## Data protection

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

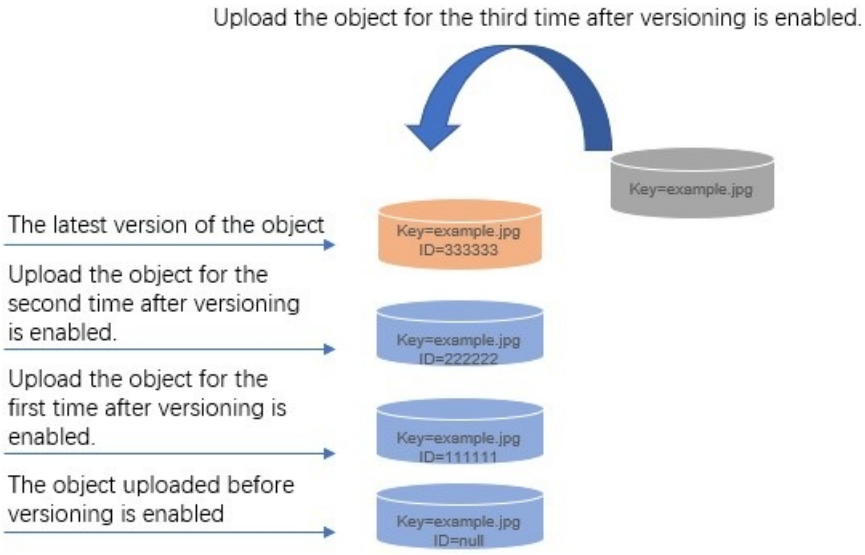
| Versioning state | Object overwriting | Object deletion |
| --- | --- | --- |
| Disabled | The existing object is overwritten and cannot be recovered. Only the current object version is retained in the bucket. | The object is deleted and cannot be accessed. |

| Versioning state | Object overwriting | Object deletion |
| --- | --- | --- |
| Enabled | A new version with a unique ID is generated for the object. The existing object is stored as a previous version. | A delete marker with a globally unique version ID is added to the object as the current version. The existing object is stored as a previous version. |
| Suspended | A new version whose version ID is null is generated for the object.<br>If the object already has a previous version or delete marker whose version ID is null, the previous version or delete marker is overwritten by the new null version. Other previous versions or delete markers whose version IDs are not null are not affected. | A delete marker whose version ID is null is added to the object.<br>If the object already has a previous version or delete marker whose version ID is null, the previous version or delete marker is overwritten by the new delete marker. Other previous versions or delete markers whose version IDs are not null are not affected. |

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

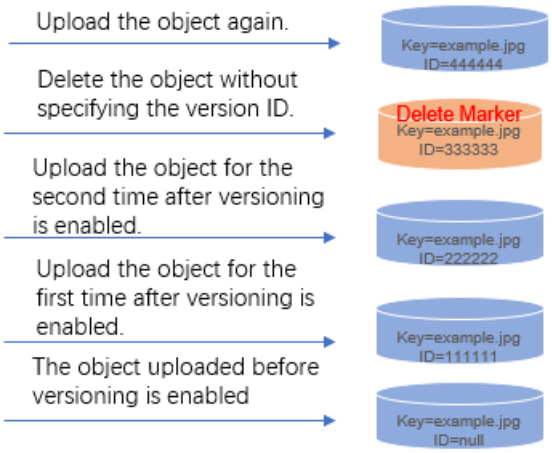
- Overwrite an object in a versioning-enabled bucket

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



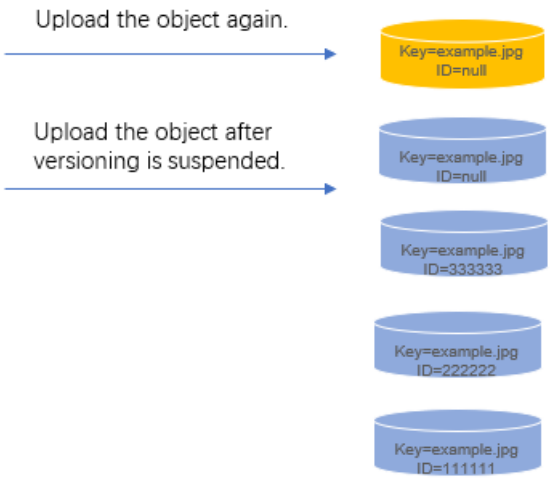

- Delete an object from a versioning-enabled bucket

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



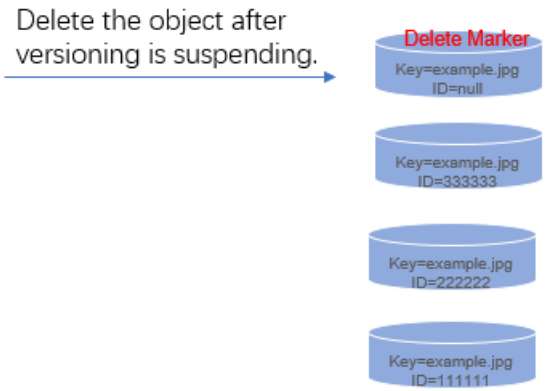

- Overwrite an object in a versioning-suspended bucket

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




- Delete an object from a versioning-suspended bucket

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




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

# 9.1.2.1.5.4. Data processing

**IMG**

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

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

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

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

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

Video snapshots

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

## Usage notes

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

## Parameters

Operation type: `video`

Operation name: snapshot

| Parameter | Description | Valid value |
| --- | --- | --- |
| t | Specifies the time when the image is to be captured. | [0, video duration]<br>Unit: ms |
| w | Specifies the width based on which to capture the image. If this parameter is set to 0, the width of the image to capture is automatically calculated. | [0, video width]<br>Unit: px |

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

## Examples

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

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

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

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

# 9.1.2.1.6. 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. Various devices, websites, and mobile applications can directly write data to and read data from OSS. You can write data to OSS by uploading files or using streams.

## Offline data storage

OSS is a cost-effective storage service that offers high data availability. You can use OSS to store enterprise data that needs to be archived offline for a long period of time.

## Cross-region disaster recovery

You can use cross-region replication (CRR) or cross-cloud replication to asynchronously replicate your data between two clusters or clouds in near real time. This way, you can build a storage architecture with three data centers in two regions and store your data in different regions for backup and disaster recovery. This ensures the continuity of your business even in case of severe disaster events.

## 9.1.2.1.7. Limits

This topic describes the limits and performance metrics of OSS.

| Item | Limit |
| --- | --- |
| Bucket | • You can create a maximum of 100 buckets.<br>• 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.<br>• If you upload an object that has the same name of an existing object in OSS, the new object will overwrite the existing object.<br>• OSS traffic is forwarded through SLB and has the following limits:<br>  ○ By default, a virtual IP address (VIP) is configured for SLB and the maximum throughput for OSS is 1.25 GB/s.<br>  ○ 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.<br>• 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. |

# 9.1.3. NAS

## 9.1.3.1. Product Introduction

### 9.1.3.1.1. What is NAS?

Apsara File Storage NAS is a cloud service that provides a file storage solution for compute nodes. The compute nodes include Elastic Compute Service (ECS) instances, Elastic High-Performance Computing (E-HPC) instances, and Container Service for Kubernetes (ACK) clusters. NAS is a distributed file storage solution that provides shared access, scalability, high reliability, and high performance.

After you create a NAS file system and a mount target, you can mount the file system on compute nodes such as ECS instances or ACK clusters. NAS allows you to access the file system by using the Network File System (NFS) protocol. You can also call POSIX-based APIs to access the file system. To share files and folders, you can mount each file system on multiple compute nodes. NAS file systems can be deployed in a hybrid manner on Hygon servers and Intel servers in a cluster.

# 9.1.3.1.2. Benefits

Apsara File Storage NAS has the following benefits:

- Parallel shared access
- High reliability
- Auto scaling
- High performance
- Ease of use
- High chip compatibility

## Parallel shared access

A file system can be mounted on multiple compute nodes at the same time to provide shared access. This access method reduces data replication and synchronization costs.

## High reliability

NAS provides reliable data storage. Compared with self-managed file systems, NAS file systems greatly reduce maintenance costs and minimize data security risks.

## Auto scaling

NAS allows you to respond to business changes in a timely manner. You can scale the capacity of a file system based on your business requirements.

## High performance

When your data storage increases, NAS file systems provide a higher throughput to meet your demand. You do not need to purchase high-end NAS storage devices. This greatly reduces upfront costs.

## Ease of use

NAS supports the NFSv3 and NFSv4 protocols. You can access file systems by calling standard POSIX API operations, regardless of the types of compute nodes and the locations of file systems.

## High chip compatibility

NAS file systems can be deployed in a hybrid manner on Hygon servers and Intel servers in a cluster.

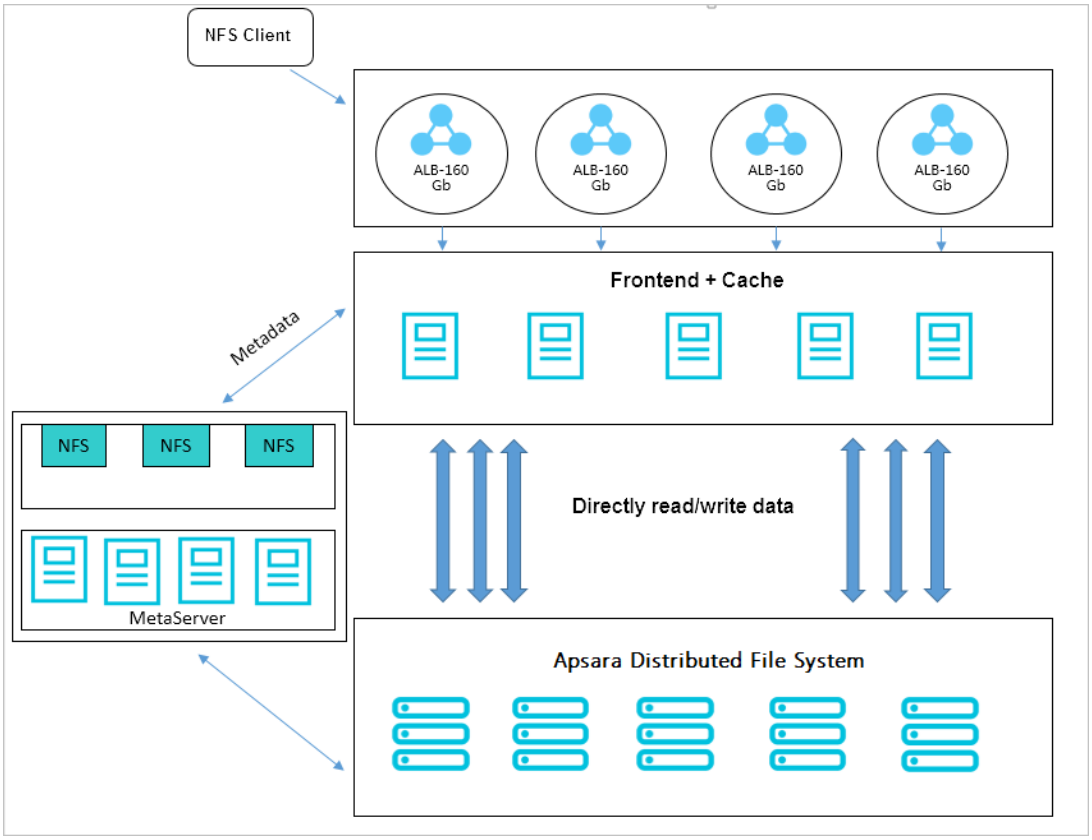
# 9.1.3.1.3. Architecture

Apsara File Storage NAS is based on Apsara Distributed File System. NAS maintains three copies for each data file across multiple storage nodes. Frontend nodes receive and cache connection requests from NFS clients. Frontend nodes are highly available because they are stateless and distributed.

The metadata of a NAS file system is stored on a MetaServer. When frontend nodes retrieve metadata from the MetaServer by using I/O requests, user data is read from and written to the backend nodes of Apsara Distributed File System.

The system architecture provides separate auto scaling of frontend and backend storage nodes. This ensures high availability, high concurrency, and low latency.

System architecture




# 9.1.3.1.4. Features

This topic describes the features of Apsara File Storage NAS.

## File system management

You can use the NAS console to manage the file systems within your Alibaba Cloud account. You can create a file system, delete a file system, and query the details of a file system.

The details page of a file system shows the basic information of the file system, such as the file system ID, region, and capacity.

### Mount target management

A mount target is an access point of a NAS file system in the classic network or a virtual private cloud (VPC). Each mount target is displayed as a domain name. You can view the mount target of a file system and modify the status and permission group of the mount target in the NAS console.

### Lifecycle management

NAS provides the lifecycle management feature. You can configure lifecycle management policies for specific directories of a file system. Then, you can dump cold data from the directories to an Infrequent Access (IA) storage medium. This way, you can manage your data by using tiered storage. To optimize the management of file storage, you can create lifecycle management policies based on your business requirements. You can view and modify lifecycle management policies in the NAS console. You can also query the storage usage of General-purpose NAS file systems and IA storage media.

### Multi-copy data replication

Data stored in NAS file systems is replicated into multiple copies. Compared with self-managed file systems, NAS file systems provide higher data reliability and lower maintenance costs and security risks.

### Unified namespace

A unified namespace contains a virtual root directory in which file systems are the first-level subdirectories. You can use a unified namespace to manage multiple file systems the same way you manage a single file system. This way, you can spend less time on data maintenance. You can create a unified namespace and a mount target for the unified namespace in the NAS console. You can also add, remove, and modify file systems in a namespace, view namespace details, and enable the cross-domain mount orchestration feature.

### Permission control and ACL-based isolation

You can configure directory-level access control lists (ACLs) for a NAS file system. You can configure ACLs for files or directories to control access by directory in a fine-grained manner. NAS allows you to control access to different directories or files of file systems by granting required permissions to users and permission groups.

### Audit logs

NAS supports the log audit feature. Logs record operations that are performed on file systems in real time. You can use the logs to analyze and identify issues.

## 9.1.3.1.5. Scenarios

This topic describes the scenarios of Apsara File Storage NAS.

### Scenario 1: shared storage and high availability for SLB

For example, assume that your Server Load Balancing (SLB) instance is connected to multiple Elastic Compute Service (ECS) instances. You can store the data of the applications on these ECS instances on a shared NAS file system. This data sharing method ensures high availability of the SLB instance.

### Scenario 2: file sharing within an enterprise

For example, the employees of an enterprise need to access the same datasets. The administrator can create a NAS file system and configure different file or directory permissions for users or user groups.

### Scenario 3: data backup

For example, you want to migrate your data from a data center to the cloud for backup. You want to use a standard interface to access the cloud storage service. You can back up your data in a NAS file system.

### Scenario 4: server logs sharing

For example, you want to store the application server logs of multiple compute nodes to a shared file store. You can store these server logs in a NAS file system for centralized log processing and analysis.

## 9.1.3.1.6. Usage notes

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

### Limits on file systems

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

### Limits on NFS clients

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

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

### Limits on SMB clients

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

## Limits on the NFS protocol

- NAS supports the NFSv3 and NFSv4 protocols.
- NFSv4.0 does not support the following attributes: FATTR4_MIMETYPE, FATTR4_QUOTA_AVAIL_HARD, FATTR4_QUOTA_AVAIL_SOFT, FATTR4_QUOTA_USED, FATTR4_TIME_BACKUP, and FATTR4_TIME_CREATE. If one of the preceding attributes is applied to a file system, an NFS4ERR_ATTRNOTSUPP error appears on the client on which the file system is mounted.
- NFSv4.1 does not support the following attributes: FATTR4_DIR_NOTIF_DELAY, FATTR4_DIR_NOTIF_DELAY, FATTR4_DACL, FATTR4_SACL, FATTR4_CHANGE_POLICY, FATTR4_FS_STATUS, FATTR4_LAYOUT_HINT, FATTR4_LAYOUT_TYPES, FATTR4_LAYOUT_ALIGNMENT, FATTR4_FS_LOCATIONS_INFO, FATTR4_MDSTHRESHOLD, FATTR4_RETENTION_GET, FATTR4_RETENTION_SET, FATTR4_RETENTEVT_GET, FATTR4_RETENTEVT_SET, FATTR4_RETENTION_HOLD, FATTR4_MODE_SET_MASKED, and FATTR4_FS_CHARSET_CAP. If one of the preceding attributes is applied to a file system, an NFS4ERR_ATTRNOTSUPP error appears on the client on which the file system is mounted.
- NFSv4 does not support the following operations: OP_DELEGPURGE, OP_DELEGRETURN, and NFS4_OP_OPENATTR. If one of the preceding operations is applied to a file system, an NFS4ERR_ATTRNOTSUPP error appears on the client on which the file system is mounted.
- NFSv4 does not support delegations.
- UID and GID
  - On Linux, mappings between UIDs or GIDs and usernames or group names are defined in configurations files. For NFSv3 file systems, if the mapping between an ID and a name is defined in a configuration file, the name is displayed. If no mapping can be found for a UID or GID, the UID or GID is displayed.
  - For NFSv4 file systems, if the version of a Linux kernel is earlier than 3.0, the usernames and group names are displayed as nobody for all files. If the kernel version is later than 3.0, the rule used by NFSv3 file systems applies to display files.

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

## Limits on the SMB protocol

- NAS supports the SMB 2.1 protocol and later versions. NAS also supports operating systems including Windows 7, Windows Server 2008 R2, and later versions. NAS does not support Windows Vista, Windows Server 2008, or earlier versions. Compared with SMB 2.1 and later versions, SMB 1.0 provides lower performance and fewer features. Windows products that support only SMB 1.0 have reached end of support.
- SMB file systems do not support extended file attributes and client-side caching based on leases.

- SMB file systems do not support I/O control (IOCTL) or file system control (FSCTL) operations. For example, you cannot create sparse files, compress files, check the status of network interface cards (NICs), or create reparse points.
- SMB file systems do not support alternate data streams.
- SMB file systems do not support identity authentication that is provided by Active Directory (AD) or Lightweight Directory Access Protocol (LDAP).
- SMB file systems do not support several features that are provided by SMB 3.0 or later, such as SMB Direct, SMB Multichannel, SMB Directory Leasing, and Persistent File Handle.
- SMB file systems do not support access control lists (ACLs) on files or directories.

### Other limits

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

## 9.1.3.1.7. Terms

This topic describes the basic terms of Apsara File Storage NAS.

### mount target

A mount target is the access address of a NAS file system in a VPC or classic network. Each mount target corresponds to a domain name. To mount a NAS file system to a local directory, you must specify the domain name of the mount target.

### permission group

The permission group mechanism is a whitelist mechanism provided by NAS. You can add rules to a permission group of a NAS file system. You can allow users from specified IP addresses or CIDR blocks to access the NAS file system by using different permissions.

> **Note** Each mount target must be associated with a permission group.

### authorized object

An authorized object is an attribute of a permission group rule. It specifies the IP address or CIDR block to which the permission group rule is applied. In a VPC, an authorized object can be a single IP address or a CIDR block. In a classic network, an authorized object must be a single IP address. In most cases, this IP address is the internal IP address of an Elastic Compute Service (ECS) instance.

# 9.1.4. EBS

## 9.1.4.1. Product Introduction

## 9.1.4.1.1. Block storage

Block storage provides Elastic Block Storage (EBS) devices based on a distributed storage architecture.

## Overview

EBS is a persistent random block storage service with low latency and high reliability and is designed for Elastic Compute Service (ECS). The EBS console provides an all-in-one, cloud-based management solution for EBS. You can use the EBS console to manage enterprise-level features and other features such as storage resource management, monitoring, performance analysis, disaster recovery, and alerting. As a centralized service platform for the entire EBS infrastructure, the EBS console informs you of resources in the cloud to best support business and optimize costs by analyzing long-term data trends.

## Service type

EBS is a block storage service. The following section describes the concepts of EBS:

EBS provides ECS instances with block-level storage that features low latency and high performance, durability, and reliability. EBS devices use the triplicate distributed storage to ensure data durability for ECS instances. 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 from the ECS instance.

## Differences among Apsara Stack storage services

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

Comparison among storage services

| Service | Benefit | Scenario |
| --- | --- | --- |
| EBS | 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 an EBS device and create file systems on it in the same way as you do with a physical disk. | EBS 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. |

| Service | Benefit | Scenario |
| --- | --- | --- |
| 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. | 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. |

# 9.1.4.1.2. Terms

This topic describes the terms used in Elastic Block Storage (EBS).

## EBS device

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

## disk

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

## snapshot

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

## async replication

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

# 9.1.4.1.3. EBS

# 9.1.4.1.3.1. Overview

Elastic Block Storage (EBS) devices are classified into the following types based on whether they can be attached to multiple Elastic Compute Service (ECS) instances:

- Cloud disks: A cloud disk can be attached to a single ECS instance that resides in the same zone as the cloud disk.
- Shared Block Storage devices: A Shared Block Storage device can be attached to up to four ECS instances that reside in the same zone as the Shared Block Storage device.

# 9.1.4.1.3.2. Cloud disks

Cloud disks are block-level storage devices provided by Apsara Stack for Elastic Compute Service (ECS) instances. Cloud disks can be classified based on their performance or purposes.

## Performance-based classification

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

> **Important**
>
> - If you deploy an Elastic Block Storage (EBS) cluster of the latest version, you can use only premium performance disks and standard performance disks.
> - If you update an EBS cluster from an earlier version to the latest version, you can continue to use ultra disks and standard SSDs in addition to newly supported premium performance disks and standard performance disks.

- Standard performance disks and premium performance disks are ideal for online transaction processing (OLTP) databases and NoSQL databases. Premium performance disks deliver up to 25,000 random IOPS for ECS instances.
- Ultra disks are ideal for medium I/O load scenarios and deliver up to 3,000 random IOPS for ECS instances.
- Standard SSDs are ideal for I/O-intensive scenarios and deliver stable and high random IOPS performance.

The following table compares the performance of different cloud disks.

| Comparison item | Premium performance disk | Standard performance disk | Standard SSD | Ultra disk |
| --- | --- | --- | --- | --- |
| Maximum capacity of a single cloud disk (GiB) | 32,768 GiB | 32,768 GiB | 32,768 GiB | 32,768 GiB |
| Maximum IOPS | 25,000 | 5,000 | 25,000 | 5,000 |
| Maximum throughput (MB/s) | 300 MB/S | 140 MB/S | 300 MB/s | 140 MB/s |
| Formula for calculating the IOPS per cloud disk | min(1,800 + 30 × Capacity, 25,000) | min(1,800 + 8 × Capacity, 5,000) | min(1,800 + 30 × Capacity, 25,000) | min(1,800 + 8 × Capacity, 5,000) |
| Formula for calculating the throughput per cloud disk | min(120 + 0.5 × Capacity, 300) | min(100 + 0.15 × Capacity, 140) | min(120 + 0.5 × Capacity, 300) | min(100 + 0.15 × Capacity, 140) |

| Comparison item | Premium performance disk | Standard performance disk | Standard SSD | Ultra disk |
| --- | --- | --- | --- | --- |
| Type name in API operations | cloud_pperf | cloud_sperf | cloud_ssd | cloud_efficiency |
| Scenario | • OLTP databases: relational databases such as MySQL, PostgreSQL, Oracle, and SQL Server databases<br>• NoSQL databases: non-relational databases such as MongoDB, HBase, and Cassandra databases<br>• Elasticsearch distributed logs: Elasticsearch, Logstash, and Kibana (ELK) log analysis | | Small and medium-sized development and test applications that require high data reliability | • Development and test applications<br>• System disks |

### Purpose-based classification

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

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

## 9.1.4.1.3.3. Shared disks

Shared disks are block-level data storage devices that feature high concurrency, high performance, and high reliability. Shared disks support concurrent reads and writes to multiple Elastic Compute Service (ECS) instances.

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

Shared disks can be classified into the following types based on their performance:

- **Shared standard SSD**: uses SSDs as the storage medium to provide stable and high performance storage that offers enhanced random I/O and data durability.
- **Shared ultra disk**: uses a hybrid SSD and HDD storage medium.

Cloud disks and shared disks can be used as data disks and up to 16 data disks can be attached to a single instance.

## 9.1.4.1.3.4. Features

Overview

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

## Read/write stability

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

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

## Backup

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

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

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

## Data encryption

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

## Elastic storage

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

## Disaster recovery

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

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

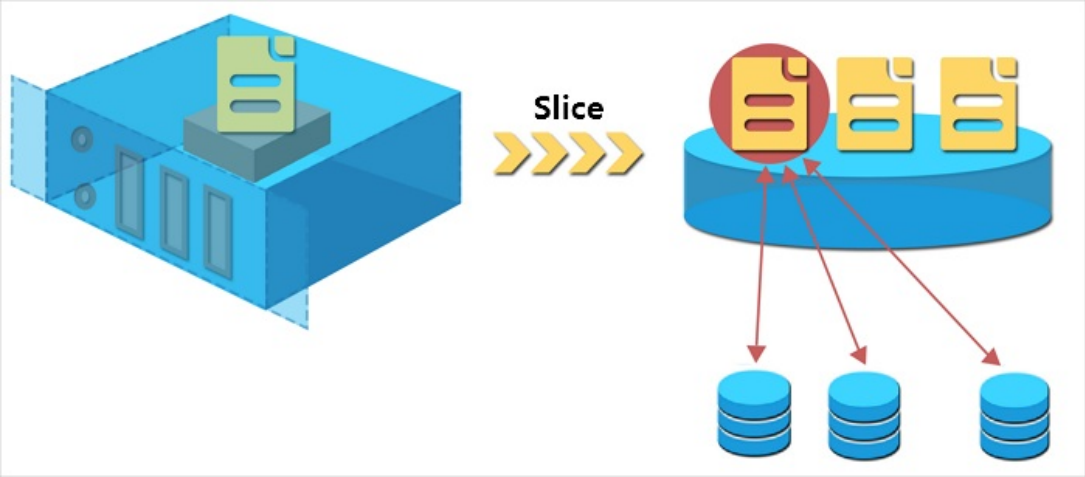
Triplicate storage

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 ensure that chunk replicas are each distributed to chunk servers on different racks and that data does not become unavailable due to the failure of a single server or rack.

## Data protection mechanism

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

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.

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)` :<br>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. |

EBS device encryption

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

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

- Data stored on EBS devices.
- Data transmitted from Elastic Compute Service (ECS) instances to EBS devices. Data within the instance operating system is not encrypted.
- All snapshots created from encrypted EBS devices. These snapshots are encrypted snapshots.

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

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

EBS device resizing

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

## Scenarios

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

- Resize an existing EBS device. You can resize the existing partitions of the EBS device or create more partitions for the EBS device.

  The following table describes the two methods for resizing an existing EBS device.

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

- Create an EBS device, attach the EBS device to the ECS instance, and then partition and format the EBS device.
- Replace the system disk of the ECS instance and specify a greater size for the new system disk.

## Maximum size of a resized system disk

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

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

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

## Maximum size of a resized data disk

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

Snapshots

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:

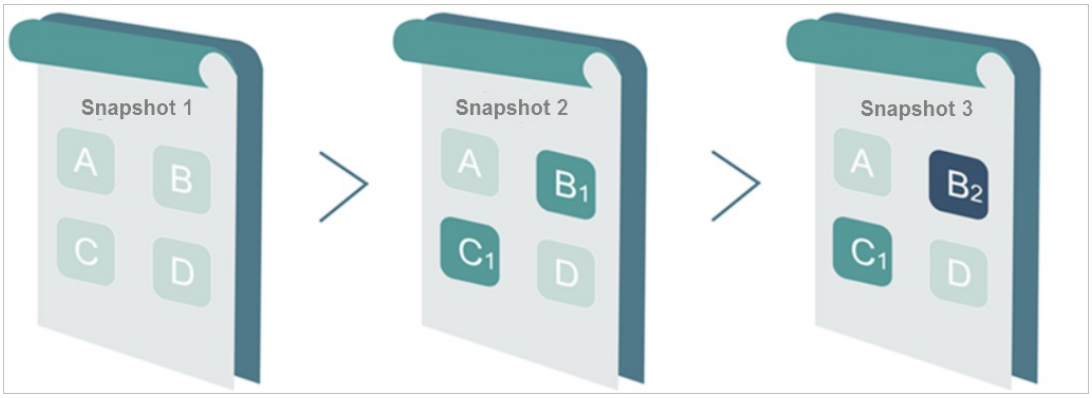
- You may want to use the data of one disk as the basis to write or store data to a different disk. To achieve this, you can create a snapshot for a cloud disk and then create another cloud disk from the snapshot. The new disk contains the basic data of the original disk.
- While cloud disks are a secure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

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.

### 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.<br>• A snapshot is created for the data disks of core business every four hours. Snapshots taken within the last ten days are retained. |
| Automatic snapshot policy | By default, the task is scheduled to be triggered once a day and cannot be modified manually. | You can customize the time of day and days of the week that snapshots are scheduled to be created and the retention period of snapshots. The disk quantity and related details associated with an automatic snapshot policy can be queried. | More flexible protection policy. | • You can schedule snapshots to be created on the hour several times in a single day.<br>• You can specify the days of the week for which to create snapshots.<br>• 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. |

### Async replication

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

## Features

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

The following table describes the operations related to async replication.

| Feature | Operation | Description |
| --- | --- | --- |
| Async replication | Create a replication pair | Before you can use the async replication feature to implement disaster recovery across zones in the same region, you must create a replication pair. |
| | Enable async replication | After you create a replication pair by specifying a primary disk and a secondary disk, you must enable the async replication feature to replicate data from the primary disk to the secondary disk across zones in the same region on a periodic basis. |
| | Disable async replication | After you enable the async replication feature, you can disable the async replication feature if you no longer require data replication or you want to perform a failover. |
| | Implement disaster recovery | After you create and activate a replication pair, if the primary disk fails, you can perform a failover and reverse replication to implement disaster recovery. |
| | Delete a replication pair | Specific limits are imposed on the disks for which you create a replication pair. Therefore, if a replication pair is no longer needed, you can delete it.<br>After the replication pair is deleted, the secondary disk rolls back to the point in time when the last async replication was complete and drops all the data that is being replicated from the primary disk. |
| | Create a replication pair-consistent group | By adding replication pairs that are used for replicating data between the primary site and the secondary site to a replication pair-consistent group, you can manage the replication pairs in a centralized manner.<br>A replication pair and a replication pair-consistent group replicate data in the same direction if they have the same primary region (production region), primary zone (production zone), secondary region (disaster recovery region), and secondary zone (disaster recovery zone). You can add replication pairs only to a replication pair-consistent group that replicates data in the same direction as the replication pairs. |

| Feature | Operation | Description |
| --- | --- | --- |
| Replication pair-consistent group | Add replication pairs | You can add replication pairs that replicate data in the same direction as a replication pair-consistent group to the group so that you can manage these replication pairs at a time by using the group. |
| | Remove replication pairs | You can remove multiple replication pairs from a replication pair-consistent group at a time. When a replication pair is removed from a replication pair-consistent group, the replication pair is disassociated from the group but is not deleted. |
| | Enable async replication for replication pair-consistent groups | After you add replication pairs to a replication pair-consistent group, you must activate the replication pairs by enabling async replication for the group to asynchronously replicate data from disks in the primary site to disks in the secondary site on a periodic basis. After replication pairs are activated, the system first performs a full synchronization to synchronize all data from disks in the primary site to disks in the secondary site. Then, the system periodically synchronizes incremental data based on the recovery point objective (RPO) of the replication pair-consistent group. |
| | Disable async replication for replication pair-consistent groups | After replication pairs in replication pair-consistent groups are activated, if you no longer require data replication or you want to perform a failover, you can disable the async replication feature for the replication pair-consistent groups. |
| | Implement disaster recovery | After you create and activate a replication pair-consistent group, if disks in the primary site fail, you can use the async replication feature to restore data on the disks by using disks in the secondary site. |
| | Delete a replication pair-consistent group | You can delete a replication pair-consistent group that you no longer need.<br>After the replication pair-consistent group is deleted, disks in the secondary site roll back to the point in time when the last async replication was complete and drop all the data that is being replicated from disks in the primary site. |

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

| Term | Description |
| --- | --- |
| async replication | The async replication feature can asynchronously replicate data from a primary disk in one zone to a secondary disk in another zone in the same region on a periodic basis. In async replication, the data on the primary disk may be inconsistent with that on the secondary disk. |
| primary disk | The disk from which you replicate data for disaster recovery. |
| secondary disk | The disk on which the replicated data is stored. |

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

## Scenarios

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

- Disaster recovery

  If the primary disk in a replication pair fails, you can perform a failover to switch the primary and secondary disks over. During the failover, the original replication link is disconnected, and services are switched over to the disaster recovery system. This is implemented by attaching the original secondary disk to the disaster recovery system.

- Cross-zone data migration in the same region

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

## Limits

The following table describes the limits on the async replication feature.

| Item | Description |
| --- | --- |
| Number of replication pairs created for a disk | 1 |
| Replication cycle | Async replication is automatically performed based on the specified RPO period. Valid values of RPO range from 2 to 10080, in minutes. |

| Item | Description |
| --- | --- |
| Type of a primary disk | A primary disk must be a data disk or system disk of the premium performance disk type. |
| Type of a secondary disk | A secondary disk must be of the same disk type and have the same performance level and capacity as the associated primary disk. |

**The following table describes the limits that apply to primary and secondary disks when you use the async replication feature.**

| Item | Method | Support for primary disks | Support for secondary disks | Description |
| --- | --- | --- | --- | --- |
| Mount a disk | Use the Elastic Compute Service (ECS) console | ✓ | ✓ | • For a secondary disk, if you attach the disk to an ECS instance when the instance is running, the operation is successful. If you attach the disk when the ECS instance stops, the ECS instance fails to start next time.<br>• If a secondary disk is in the **Attaching** state, the replication pair to which the secondary disk belongs fails to be activated. |
| Use a system disk | Use the ECS console | ✓ | ✓ | • If the system disk of an ECS instance is used as a primary disk, the ECS instance can read and write data from and to the disk as expected.<br>• If the system disk of an ECS instance is used as a secondary disk, the ECS instance may fail to start, or you may fail to access the operating system of the instance after startup. |
| Read and write data from and to disks | Use the ECS console | ✓ | × | After a replication pair is activated, the secondary disk enters the read-only state, and no users have write permissions on the secondary disk. |

| Item | Method | Support for primary disks | Support for secondary disks | Description |
| --- | --- | --- | --- | --- |
| Delete a disk | • Use the ECS console<br>• Call the DeleteDisk operation | ✓ | ✓ | After a disk is deleted for a period of time, the state of the replication pair to which the disk belongs changes to **Invalid**. |
| Initialize a disk | • Use the ECS console<br>• Call the ReinitDisk operation | ✓ | ✓ | After a disk is initialized for a period of time, the state of the replication pair to which the disk belongs changes to **Invalid**. |
| Create a snapshot | • Use the ECS console<br>• Call the CreateSnapshot operation | ✓ | ✓ | Due to the RPO setting, data of a snapshot created for a primary disk may be inconsistent with that of a snapshot created at the same time for the associated secondary disk. |
| Roll back data based on snapshots | Not supported | × | × | N/A |
| Replace a system disk | • Use the ECS console<br>• Call the ReplaceSystemDisk operation | ✓ | ✓ | After you perform this operation, the original disk is deleted. Therefore, this operation is equivalent to the DeleteDisk operation. After the system disk of an ECS instance is replaced for a period of time, the state of the replication pair to which the system disk belongs changes to **Invalid**. |
| Resize a disk online | • Use the ECS console<br>• Call the ResizeDisk operation | ✓ | ✓ | After you resize a disk, the state of the replication pair to which the disk belongs changes to **Invalid**. |

| Item | Method | Support for primary disks | Support for secondary disks | Description |
| --- | --- | --- | --- | --- |
| Resize a disk offline | • Use the ECS console<br>• Call the ResizeDisk operation | ✓ | ✓ | After you resize a disk, the state of the replication pair to which the disk belongs changes to **Invalid**.<br>**Important** After you resize a disk offline, make sure that the new disk size takes effect before you create a replication pair for the disk. Otherwise, the replication pair may become invalid. |
| Change the type of a disk | Not supported | × | × | N/A |
| Encrypt a disk | Not supported | × | × | You cannot convert a non-encrypted disk to an encrypted disk. |
| Migrate disks along with ECS instances | Not supported | × | × | N/A |

# 9.1.4.1.4. 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.

# 9.1.5. Log Service

## 9.1.5.1. Product Introduction

# 9.1.6. CSG

## 9.1.6.1. Product Introduction

### 9.1.6.1.1. What is CSG?

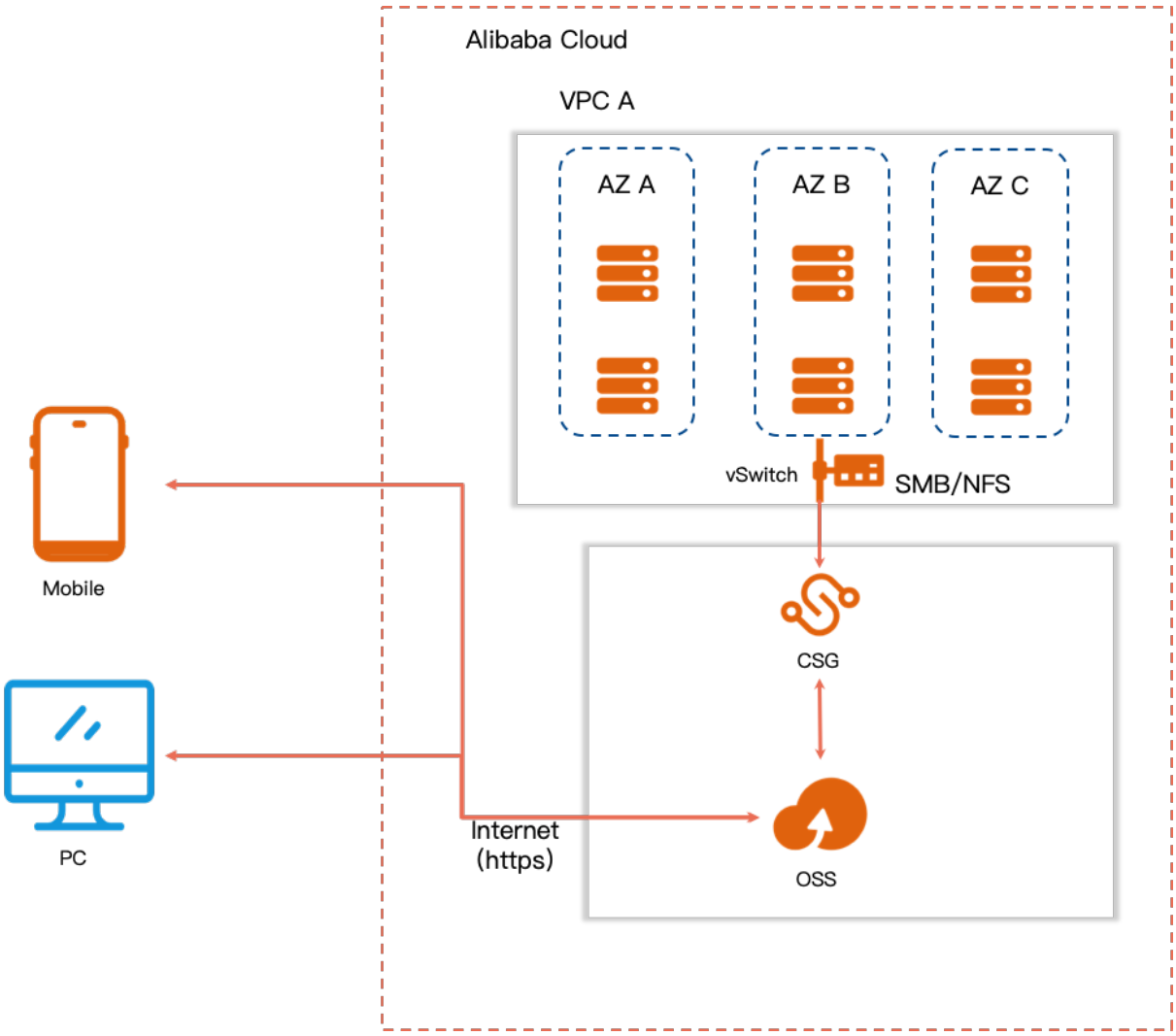
Cloud Storage Gateway (CSG) is a gateway service that can be deployed on Apsara Stack. CSG uses OSS buckets as backend storage devices. CSG provides on-premises and cloud applications with standard file services over the Network File System (NFS) and Server Message Block (SMB) protocols.

CSG supports only file gateways.

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

#### Service architecture

The following figure shows the architecture of CSG.



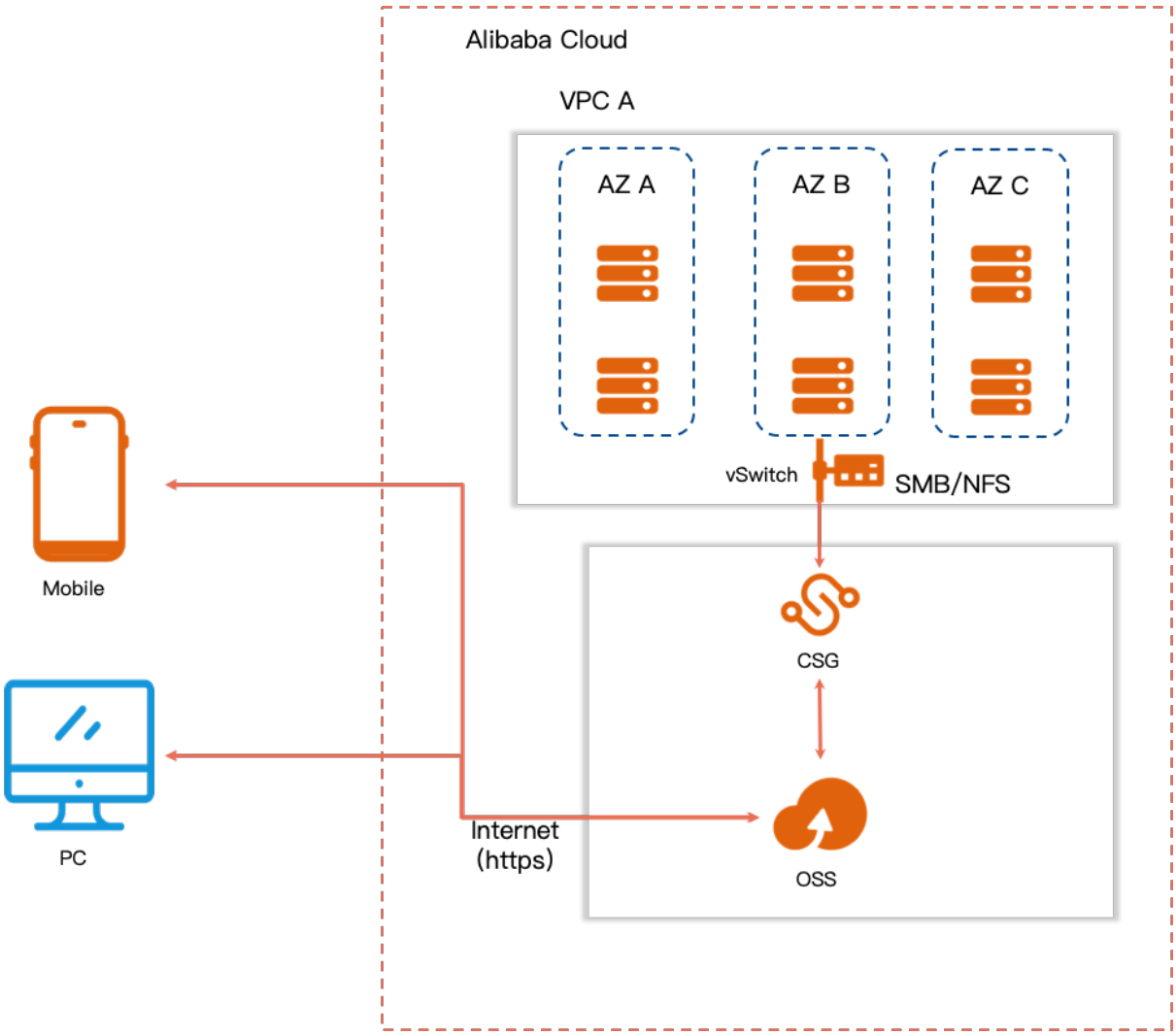

## Scenarios

File gateways

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

# 9.1.6.1.2. Architecture

The following figure shows the architecture of Cloud Storage Gateway (CSG).




# 9.1.6.1.3. Specifications

This topic describes the specifications of file gateways that are deployed in Apsara Stack.

## File gateways

The following table describes the specifications of file gateways.

Specifications of file gateways deployed in Apsara Stack

| Item | Standard |
| --- | --- |
| Storage protocol | NFSv3, NFSv4.0, and SMB |
| Maximum number of files (recommended) | 50 million |
| Maximum capacity of shared file systems (recommended) | 256 TB |
| Maximum number of shared files | NFS: 8, SMB: 8 |

| Item | Standard |
| --- | --- |
| Maximum number of agents concurrently connected to a gateway | NFS: 1,024, SMB: 1,024 |
| Maximum bandwidth for OSS-based synchronization | 2 Gbps |
| Encrypted transmission to OSS | Yes |
| Cache capacity | 40 GB~2 TB |
| Maximum gateway read bandwidth | 2.5 Gbps |
| Maximum gateway write bandwidth | 2 Gbps |

# 9.1.6.1.4. Features

This topic describes the features of standard file gateways.

## File gateways

File gateways support two network transmission protocols: Network File System (NFS) and Server Message Block (SMB). You can use both protocols to transfer files over LAN.

- NFS allows you to access file systems in UNIX, such as AIX, HP-UX, and other Linux-based operating systems.
- SMB allows you to access file systems in Windows.

Features of file gateways

| Feature | | Standard | References |
| --- | --- | --- | --- |
| Basic features | Cache disks | √ | N/A |
| | NFS and CIFS services | √ | N/A |
| Operations | Modify gateways | √ | N/A |
| | Upgrade gateways | √ | N/A |
| Advanced features | Reverse synchronization | √ | N/A |
| | Intelligent caching | √ | N/A |
| | Intelligent replication | √ | N/A |
| | Ignore deletion | √ | N/A |

> **Note** In the preceding table, the check sign (✓) indicates that the feature is supported, and the cross sign (×) indicates that the feature is not supported.

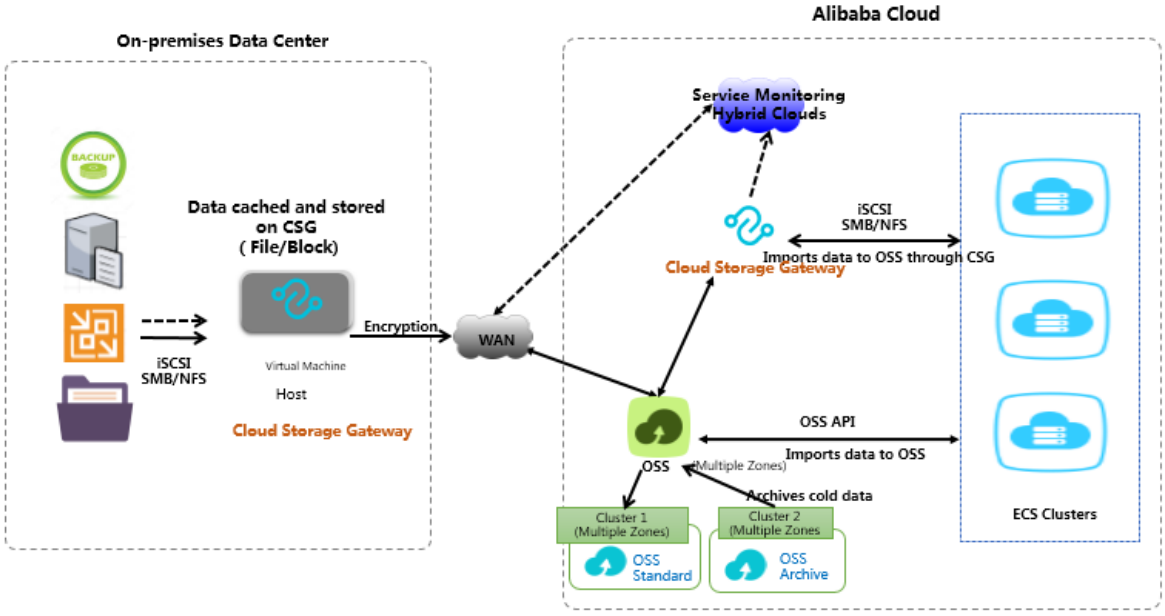
# 9.1.6.1.5. Scenarios

This topic describes the use scenarios of Cloud Storage Gateway (CSG).

## Storage expansion and data migration

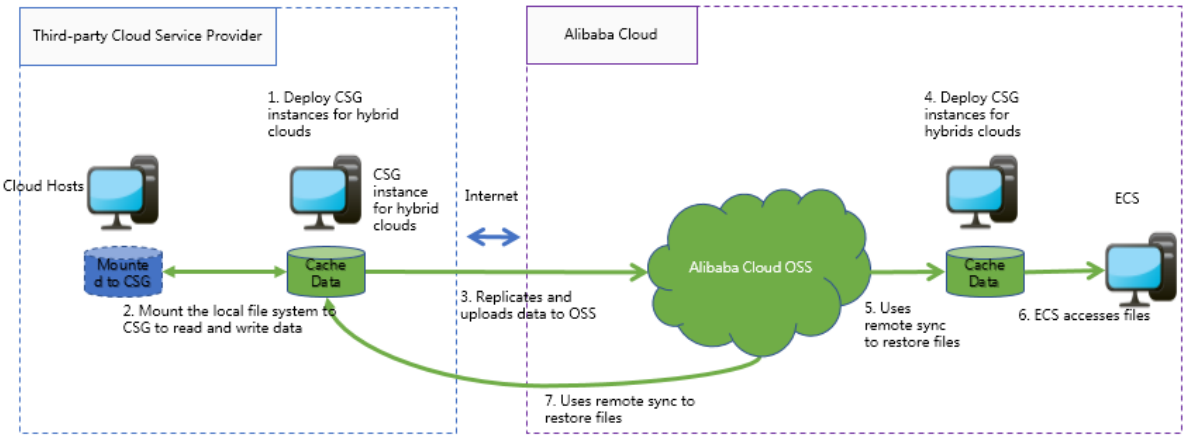
CSG is integrated with intelligent caching algorithms, which automatically distinguish between hot and cold data. This technology allows CSG to store cold data in the cloud and retain hot data in local cache disks to accelerate data retrieval. You can transfer a large amount of data in the cloud storage to a data center without additional configurations. This way, you can expand your storage from a data center to the cloud. The cloud also retains a full copy of data. This ensures the integrity of both your hot and cold data. CSG can be used in the following scenarios:

- File sharing: Files and data are shared among different computing clusters.
- Data backup: Application data is synchronized to Alibaba Cloud Object Storage Service (OSS) by using CSG and backup software, such as Veeam and NBU.



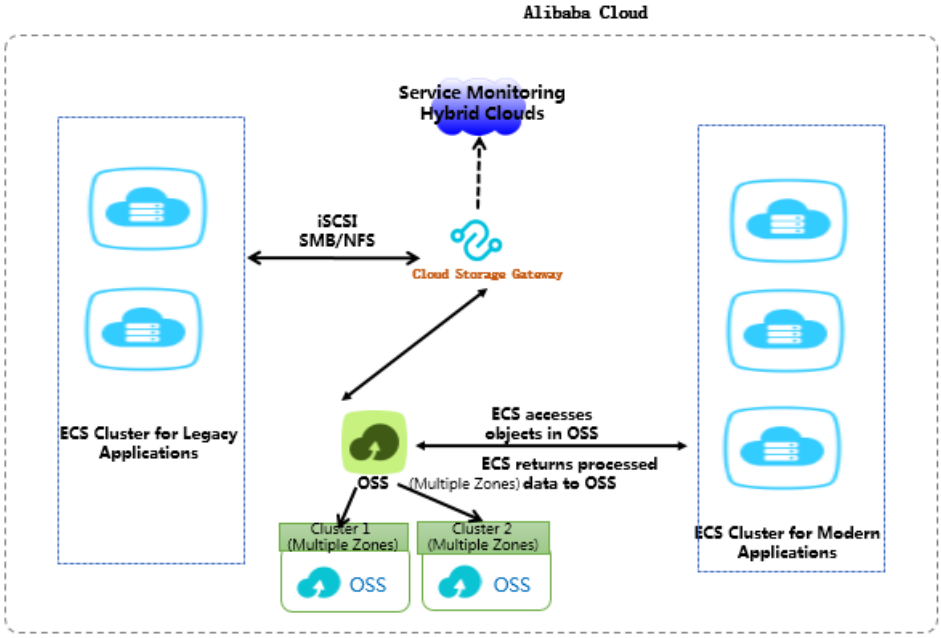

## Cloud disaster recovery

As cloud computing becomes more and more widely used, an increasing number of users migrate their services to the cloud. Therefore, users are facing the challenge on the reliability and continuity of services that are running in the cloud. CSG supports virtualization. You can integrate third-party services with Alibaba Cloud services to implement disaster recovery.




## Compatibility with legacy applications

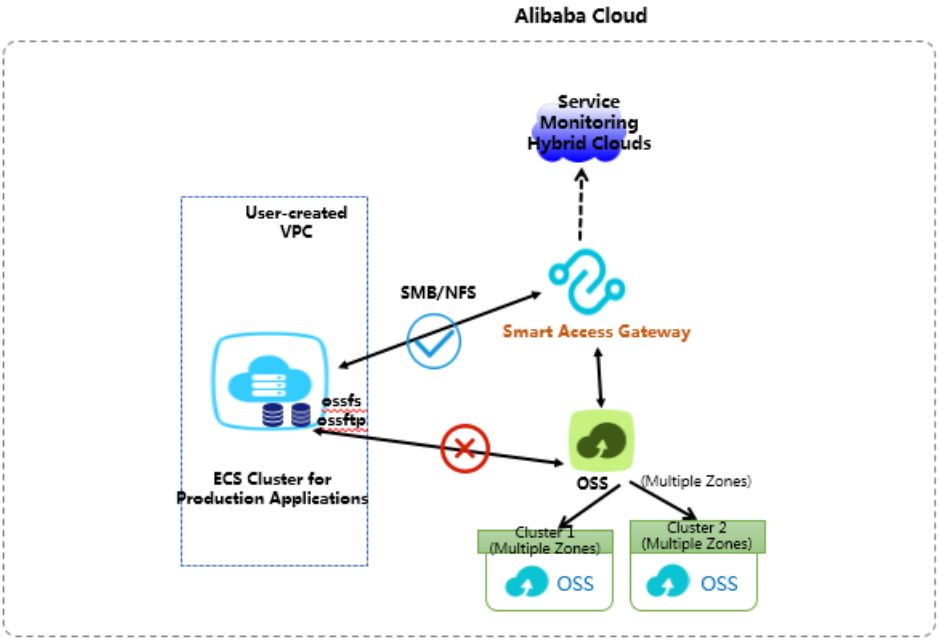
Some users may run both legacy and modern applications in the cloud. In this case, the legacy applications that are migrated from a data center use standard storage protocols, such as NFS and SMB. In most cases, modern applications are developed based on cutting-edge technologies and support object access protocols. Data communication among applications over different protocols requires complicated operations. CSG allows you to establish communications among applications over different protocols and enable data exchange between legacy applications and modern applications.




## An alternative to ossfs and ossftp

ossfs and ossftp are open source tools based on file protocols. You can use ossfs and ossftp to directly upload files to OSS. However, ossfs and ossftp are not supported in the production environment due to their low compatibility with POSIX. Additional configurations and cache disks are required to use ossfs and ossftp to mount file systems to a client. In scenarios where you need to use ossfs and ossftp to mount file systems on multiple clients, the configuration process may require a long period of time.

CSG is an excellent alternative to ossfs and ossftp. To accelerate access to data that is stored in OSS, you only need to create a CSG instance, and mount an NFS share on your local client or map a Windows SMB share to a network drive. Then, you can manage data in the remote OSS bucket in the same way that you manage data in a local file system.




# 9.1.6.1.6. Usage notes

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

## File gateways

- You can create file gateways only by using the recommended CIDR block or its subnets of a VPC.
- You may have a large number of files exist in the same directory of an OSS bucket. For example, hundreds of thousands or even millions of subdirectories or files exist in the root directory or under a specific directory of an OSS bucket. In this case, file gateways are not suitable due to extremely slow access to subdirectories or files. In most cases, if you want to use file gateways to manage files, we recommend that you place at most 100,000 files in a single directory.
- Files gateways reserve 20% of the cache disk space to cache metadata and use the rest of the space to cache data. The stored metadata refers to the metadata of different files, including the size and last modification time of each file. In this case, if a file is discarded, a file gateway creates a metadata file so that you can view the information about the discarded file when you access folders from the client. If you want to read the discarded file, the file gateway retrieves the data of the discarded file from the OSS bucket to the cache disk. The space required to cache the metadata on the cache disk determines the maximum number of files that are supported by the current share. In most cases, a share with a cache disk of 100 GB in size supports at most 10 million files. You can check the value of the Available Metadata Space parameter of shares when you use the file gateway. Generally, you can consider whether to scale out the cache disk when the remaining space is about 3 GB.

- Files written to file gateways are asynchronously synchronized to OSS buckets. Before the files are synchronized to OSS buckets by using file gateways, we recommend that you do not modify or delete the files by using other gateways or tools, such as ossutil or ossfs. Otherwise, data inconsistency may occur.
- We recommend that you do not frequently interrupt the upload of large files to NFS or SMB shares. The system uploads files by using multipart uploads. If you interrupt the upload of large files, file fragments are generated in the associated OSS bucket. These file fragments occupy the capacity of the OSS bucket. Therefore, the storage usage of the OSS bucket is slightly higher than the total file size.
- The cache capacity for file sharing is calculated based on the following formula: Recommended local cache capacity = [Application bandwidth (Mbit/s) - Backend bandwidth of a gateway (Mbit/s)] × Write duration (seconds) × 1.2.

  To obtain better performance when you access data from local clients, you can estimate the total amount of hot data. Compare the total amount of hot data with the recommended on-premises cache capacity and select the higher value as the capacity of the on-premises cache disk.
- If you want to write large files by using a file gateway, the size of each file must be smaller than 30% of the cache disk capacity. You cannot write multiple large files at the same time. If you write multiple large files at the same time, the cache disk space is exhausted.
- File gateways support sparse files. If a sparse file fails to be uploaded to a file gateway, run the following command to convert the format of the sparse file:

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

  The size of the sparse file cannot exceed the available capacity of the cache disk.
- The names of file gateways and directories must be encoded in UTF-8. File gateways only support file and directory names that are encoded in UTF-8. Other formats are not supported. For example, if you mount an NFS file share of a file gateway on a Windows CSG agent, the files and directories that have Chinese names cannot be created. In this case, a 0x8007045D error is returned.
- If the size of a file in a file gateway exceeds 256 MB, we recommend that you disable the versioning feature for the associated OSS bucket. Otherwise, a timeout error may occur when the gateway uploads metadata to the associated bucket. This affects the performance of the gateway.

# 9.1.6.1.7. Reliability and consistency

This topic describes the reliability and consistency of Cloud Storage Gateway (CSG).

## Reliability

- In cache mode, data is written to a disk by using synchronous I/O to prevent data loss due to power failures.
- CSG ensures the durability and reliability of data in the local cache disks by using Alibaba Cloud disks.
- CSG refreshes and uploads data in the local cache disks to an OSS bucket. Alibaba Cloud OSS guarantees 99.9999999999% reliability. This ensures data security and reliability when you

transfer data from the gateway to Alibaba Cloud.

## Consistency

- A 64-bit cyclic redundancy check (CRC-64) is implemented on the data that is uploaded to an OSS bucket from the gateway. If you send data from your data center to an OSS bucket, a checksum is generated and sent along with the data. Another checksum is generated by OSS and compared with the checksum that is received by OSS. If the two are equal, the uploading process is complete.
- If you upload a small file to an OSS bucket, the file gateway generates a checksum of the file. If you upload a large file in multiple parts to an OSS bucket, the file gateway implements CRC-64 on each part. This ensures consistency between the two checksums that are generated in your data center and the OSS bucket.

# 10.Tablestore

# 10.1. Product Introduction

## 10.1.1. What is Tablestore?

Tablestore is a NoSQL data storage 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. Table-level changes are not required to add or delete attribute columns. You can configure the time to live (TTL) parameter for a table to manage the lifecycle of data. Expired data is automatically deleted from the table.
- 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 business operations.
- Adopts the triplicate technology to keep three copies of data on different racks. A cluster can support single storage type instances (SSD only) or mixed storage type instances (SSD and HDD) to meet different budget and performance requirements.
- Adopts a fully redundant architecture that prevents single points of failure (SPOFs). Tablestore supports smooth online upgrades, hot cluster upgrades, and automatic data migration, which enable you to dynamically add or remove nodes for maintenance without incurring service interruptions. The concurrent read and write throughput and storage capacity can be linearly scaled. Each cluster can have at least 500 servers.
- Supports highly concurrent read and write operations. Concurrent read and write capabilities can be scaled out as the number of servers increases. The read and 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.

## 10.1.2. Benefits

Tablestore provides the following benefits.

### Scalability

- Tablestore does not impose any limits on the amount of data that can be stored in tables. As data increases, Tablestore adjusts data partitions to provide more storage space for tables and improve the capability of handling sudden spikes of access requests.
- 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

High-performance Tablestore instances provide single-digit millisecond latency when you access single rows of data. 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 copies of data and restores data when any of the copies become damaged.
- Tablestore supports automatic fault tolerance for server disk failures in a cluster and supports hot swapping of disks. In the event of a disk failure, services can be restored within a minute.
- Tablestore supports full and incremental backup and data restoration from storage.
- Tablestore supports the backup between data clusters in different data centers. You can view and manage the backup process.
- 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, providing 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 use Log Service to store and download audit logs. This allows you to store audit logs for extended periods of time and simplify the management of 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 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 regular users from querying access control details and simplify access control for administrators. This improves the usability of access control.

## Strong consistency

Tablestore ensures strong consistency for data writes. After three replicas are written to disks, the write operation is successful. Applications can immediately read the latest data.

## Flexible data models

Tablestore tables do not require a rigid schema. 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 number of requests per second and the average response latency.

### Multitenancy

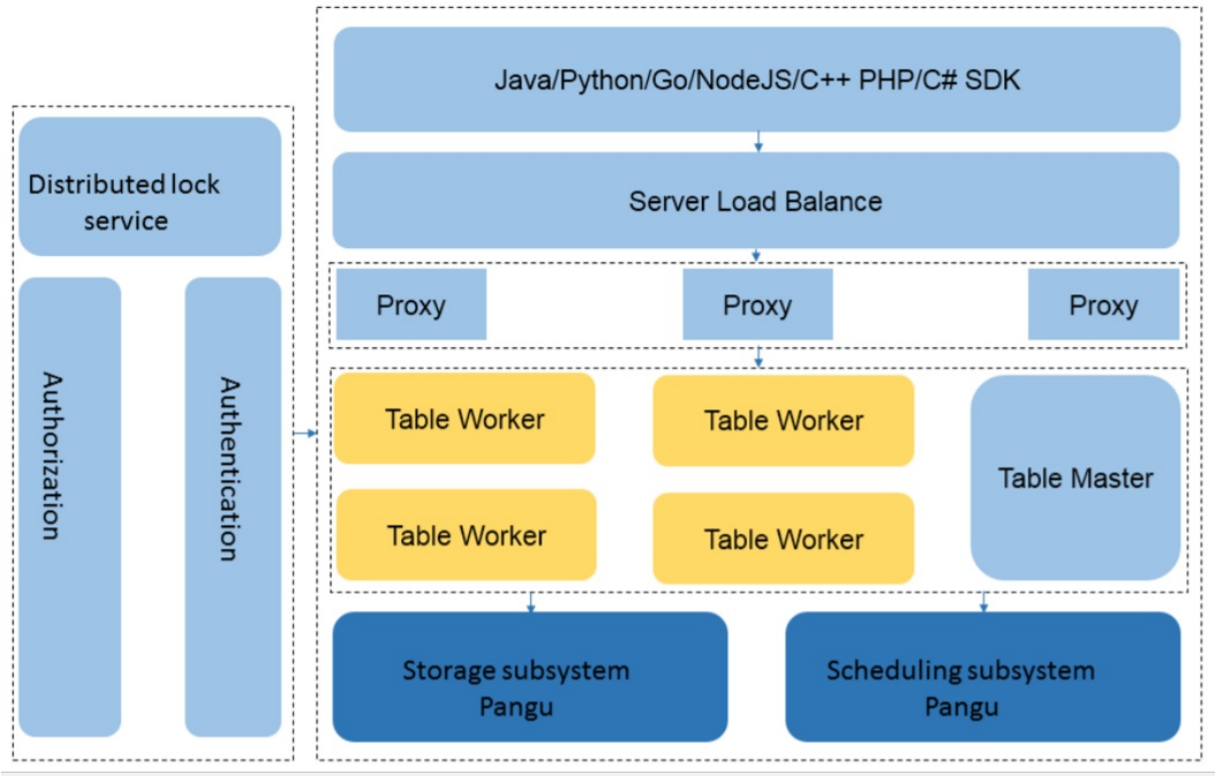
- Isolation: allows tasks of multiple tenants to be submitted to different queues and run separately. Resources are isolated among tenants.
- Permission: allows you to manage tenants in a centralized manner, dynamically configure and manage tenant resources, isolate resources, view statistics for resource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

## 10.1.3. Architecture

This topic describes the Tablestore architecture.

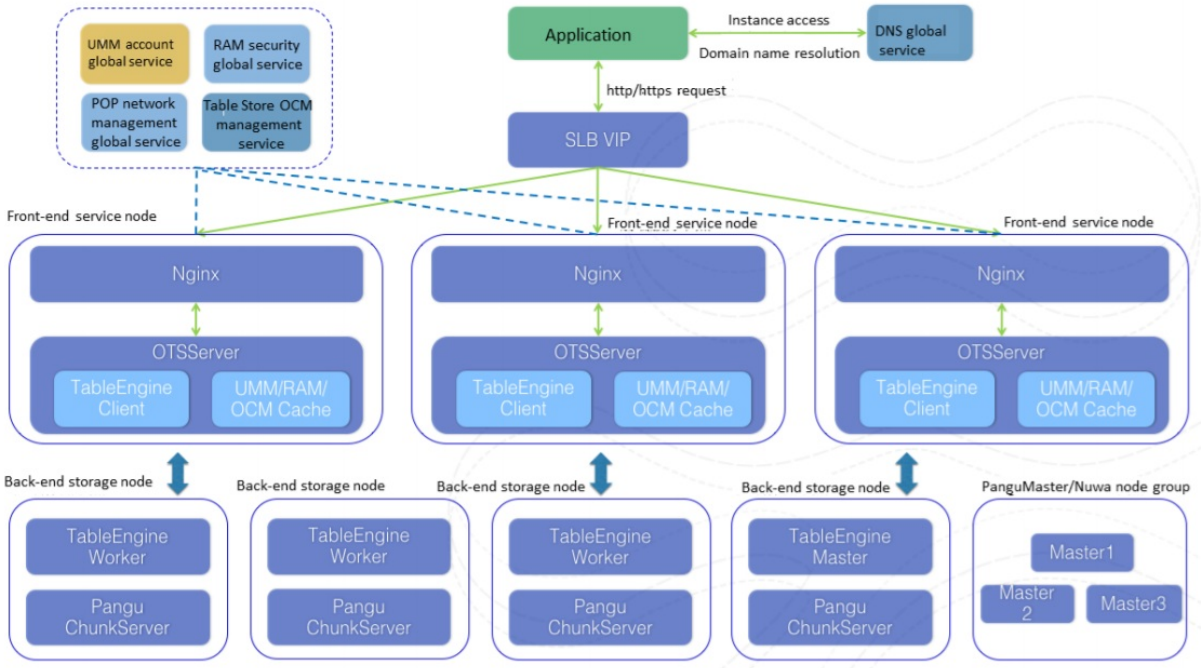
The architecture of Tablestore is referenced from Bigtable (one of the three core technologies of Google) and uses the log-structured merge-tree (LSM) storage engine to provide high write performance. The performance of primary key-based single-row queries and range queries is stable and predictable. 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.




# 10.1.4. Scenarios

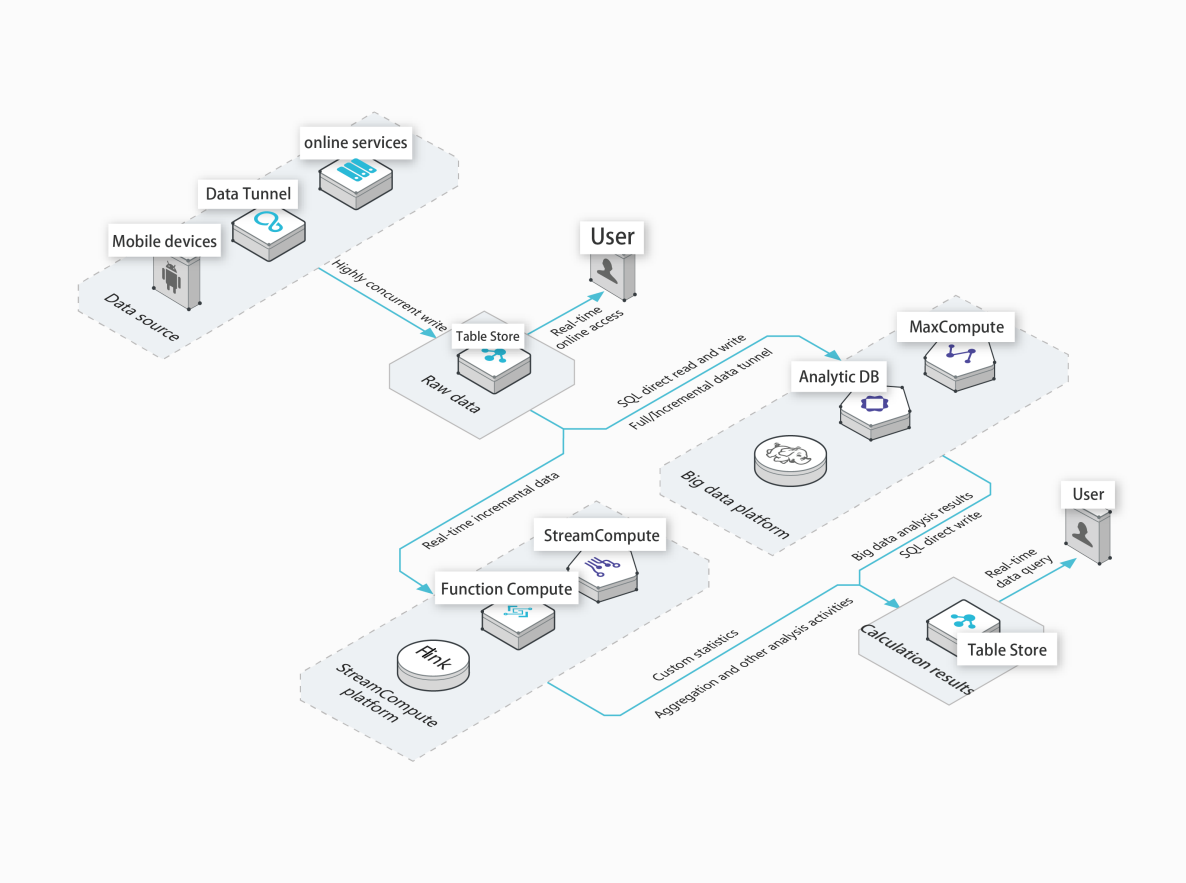
Tablestore can be applied to the following scenarios:

- 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.

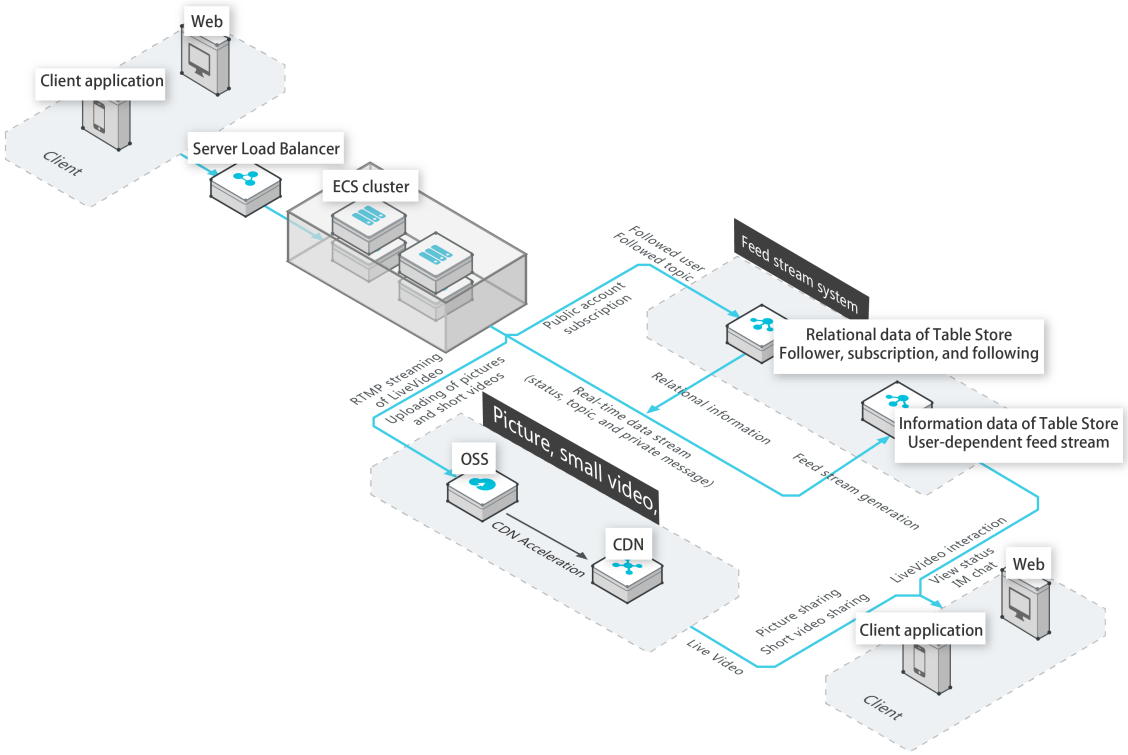



- Scenario 2: Social media feeds on the Internet

  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.

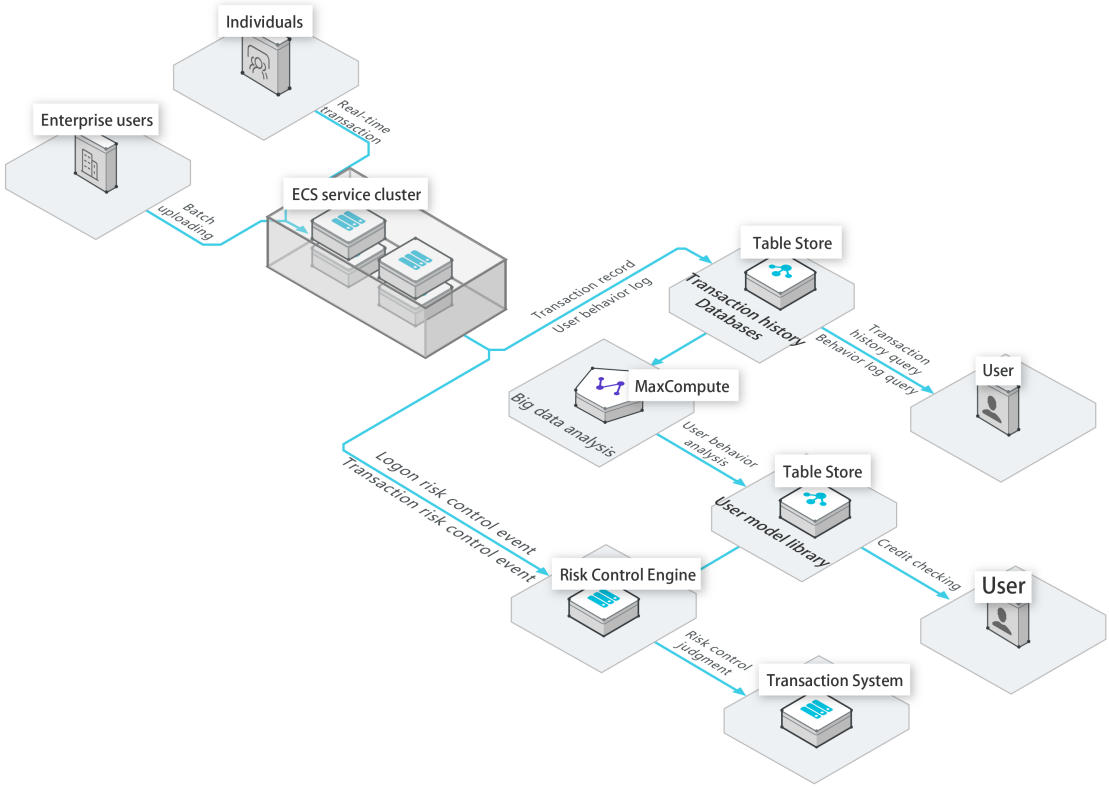



- Scenario 3: Storage and real-time queries of large amounts of transaction records and user models

  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.

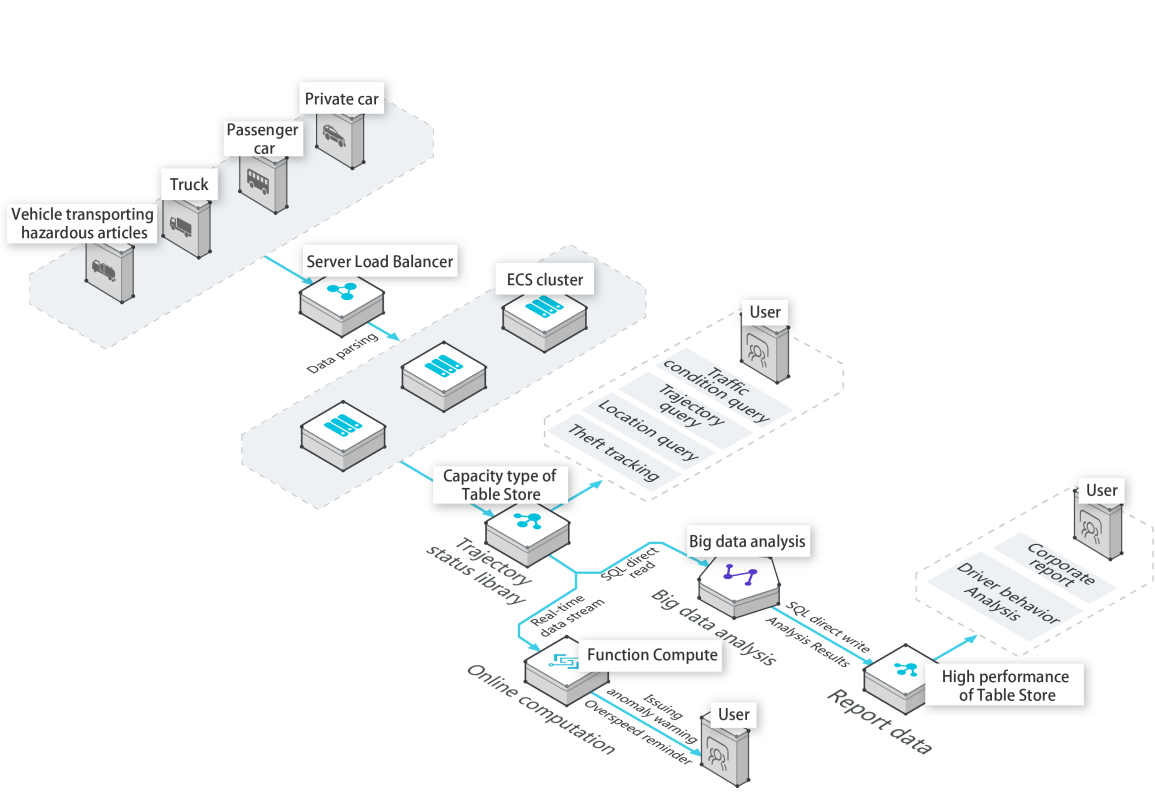



- Scenario 4: Efficient and flexible storage of large amounts of IoV data

  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.

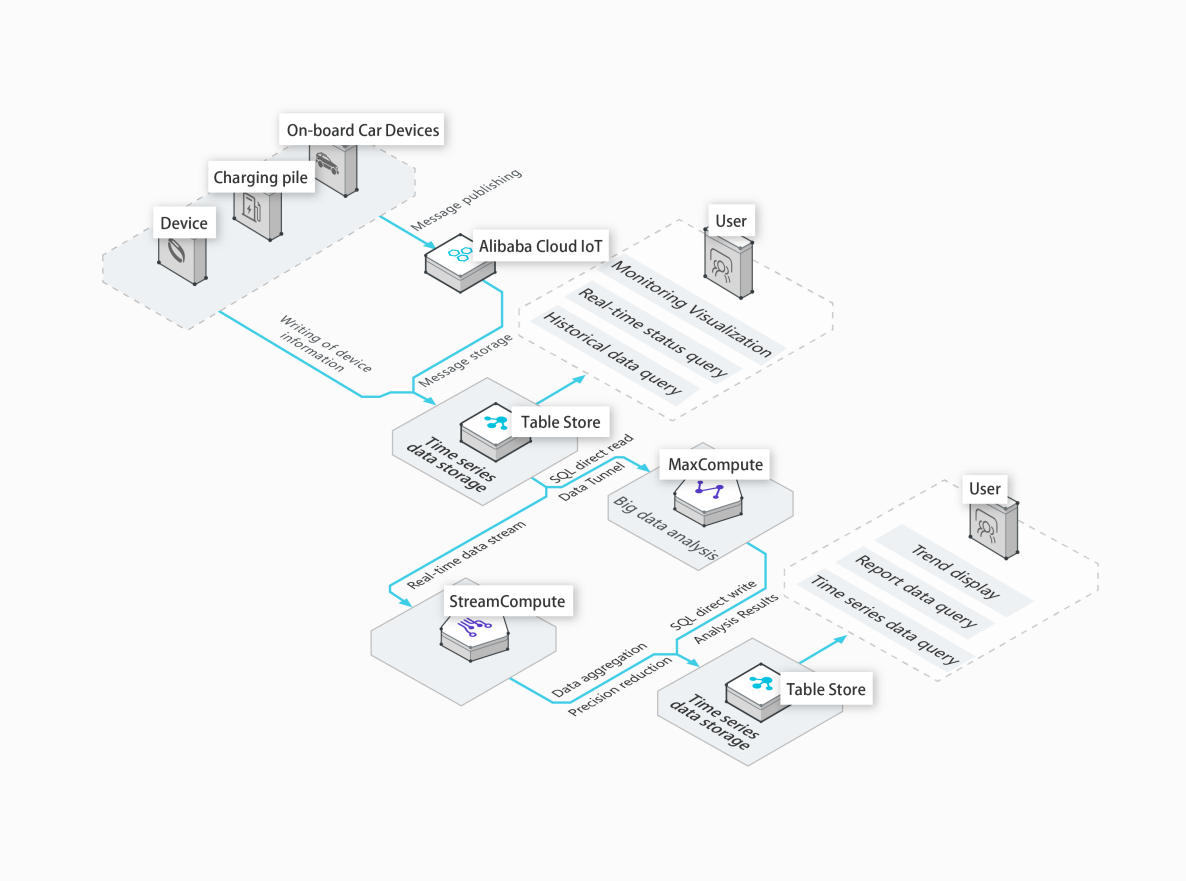



- Scenario 5: Storage of large amounts of IoT data for efficient queries and analysis

  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.

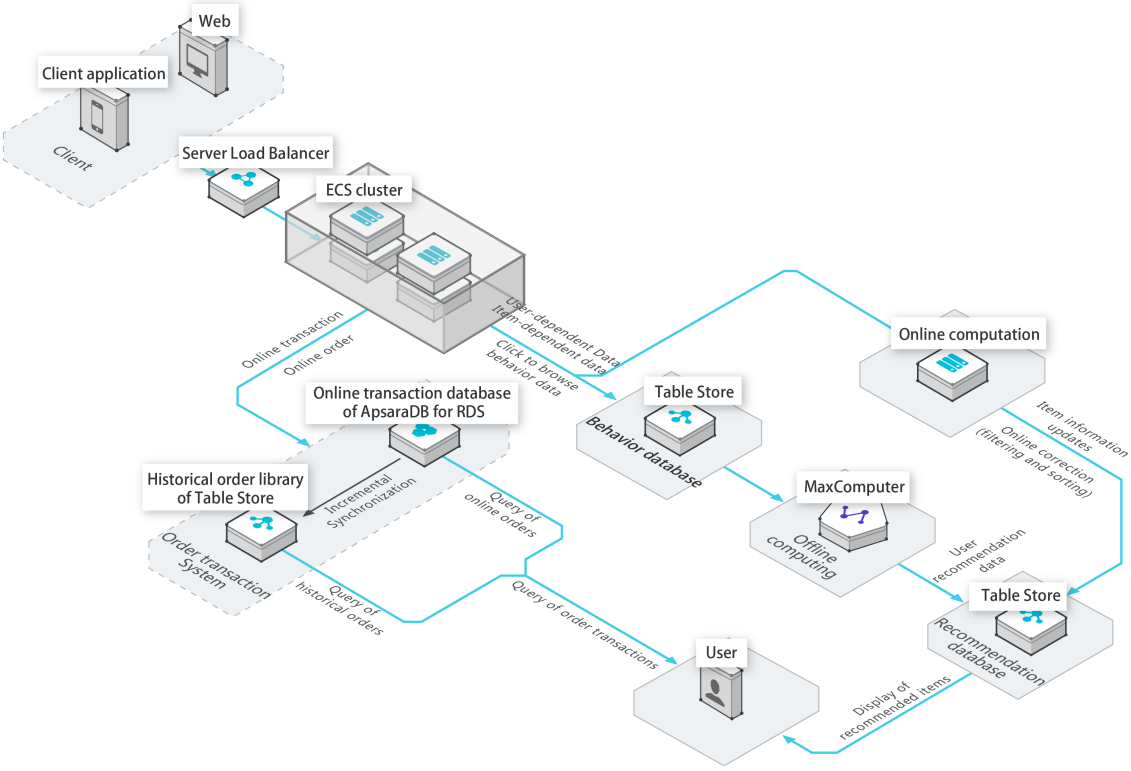



- Scenario 6: Databases for large-scale e-commerce transaction orders and user-specific recommendations

  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.




# 10.1.5. 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 length of an instance name | 3 to 16 bytes | The instance name can contain uppercase and lowercase letters, digits, and hyphens (-). It must start with a letter and cannot end with a hyphen (-). |
| The length of a table name | 1 to 255 bytes | The table name can contain uppercase and lowercase letters, digits, and underscores (_). It must start with a letter or underscore (_). |

| Item | Limit | Description |
| --- | --- | --- |
| The length of a column name | 1 to 255 bytes | The column name can contain uppercase and lowercase letters, digits, and underscores (_). It must start with a letter or underscore (_). |
| The number of columns in a primary key | 1 to 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. |
| 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 to 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<br>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. |

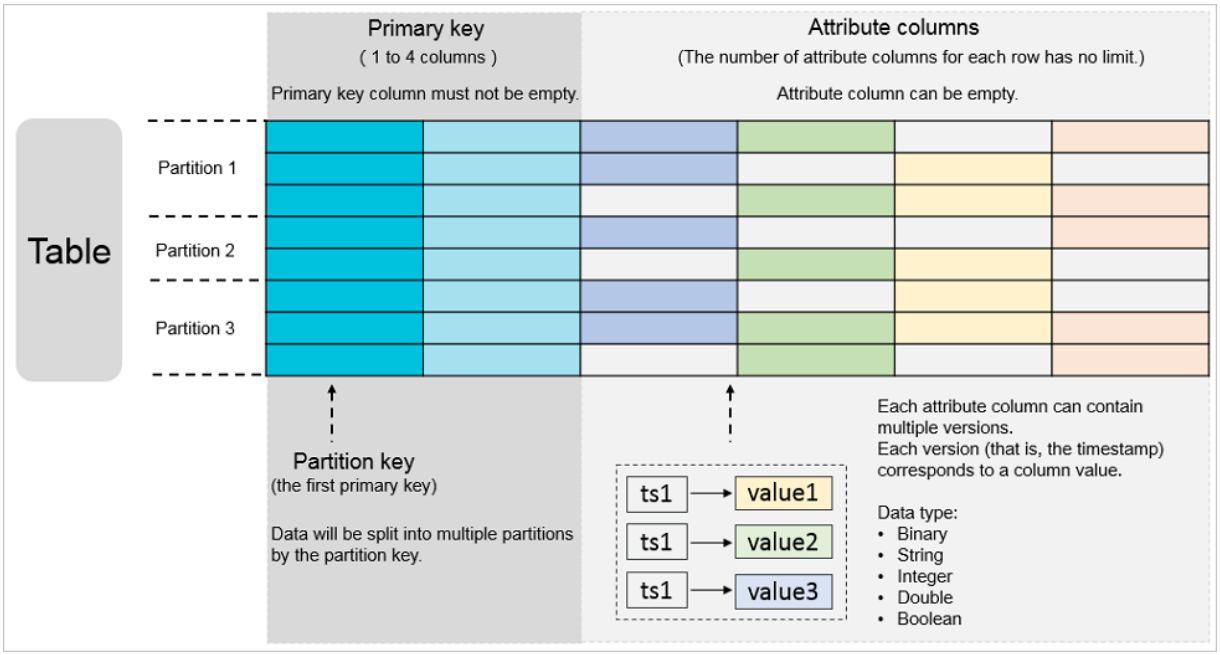
| Item | Limit | Description |
| --- | --- | --- |
| Data returned by one GetRange request | 5,000 rows or 4 MB | The amount of data returned by a request cannot exceed 5,000 rows or 4 MB. When either of the limits is exceeded, data that exceeds the limits is truncated at the row-level. The data primary key information in the next row is returned. |
| The data size of an HTTP request body | 5 MB | None. |

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

### primary key and attribute

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

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

### read/write throughput

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

### region

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

### instance

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

### endpoint

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

### SATA

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

# 10.1.7. Features

# 10.1.7.1. 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.

# 10.1.7.2. Tunnel Service

Tunnel Service is built on the Tablestore API to provide tunnels that are used to consume data in full, incremental, and differential modes. You can create full, incremental, and differential tunnels and consume distributed data through these tunnels in real time. After you create a tunnel for a data table, you can use the tunnel to consume historical and incremental data in the data table.

## Background information

Tablestore is applicable to scenarios such as metadata management, time series data monitoring, and message systems. In these scenarios, incremental or full and incremental data streams are generally used to trigger the following operations:

- Data synchronization: synchronizes data to a cache, search engine, or data warehouse.
- Event triggering: triggers Function Compute, sends a notification when data is consumed, or calls an API operation.
- Stream data processing: connects to a stream-processing engine or a unified stream- and batch-processing engine.
- Data migration: backs up data to OSS or migrates data to a Tablestore capacity instance.

## Features

The following table describes the features of Tunnel Service.

| Feature | Description |
| --- | --- |
| Tunnels for full and incremental data consumption | Tunnel Service allows you to consume incremental data, full data, and full and incremental data simultaneously. |
| Orderly incremental data consumption | Tunnel Service sequentially distributes incremental data to one or more logical partitions based on the write time. Data in different partitions can be consumed simultaneously. |
| Consumption latency monitoring | Tunnel Service allows you to call the DescribeTunnel operation to view the recovery point objective (RPO) information of the consumed data on each client. You can use the Tablestore console to monitor data that is consumed through tunnels. |
| Horizontal scaling of data consumption capabilities | Tunnel Service supports automatic load balancing among logical partitions to accelerate data consumption. |

# 10.1.7.3. Global secondary index

# 10.1.7.3.1. 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 Define0 and PK1 and specify Defined1 as an attribute column.
  - You can create an index table on Define1 and Define0 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.

# 10.1.7.3.2. 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 20 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.

# 10.1.7.3.3. 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(ce
llNumber));
    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(cell
Number));
    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(rangeRowQueryCri
teria));
        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 |
| 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(cel
lNumber));
    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(cell
Number));
    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(rangeRowQueryCri
teria));
        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 |

| PK0 | PK1 | PK2 |
| --- | --- | --- |
| 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(ba
seStationNumber));
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(st
artTime));
    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(bas
eStationNumber));
    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 star
t from" + strStartTime + "are listed:");
    while (true) {
        GetRangeResponse getRangeResponse = client.getRange(new GetRangeRequest(rangeRowQueryCri
teria));
        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(ba
seStationNumber));
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(st
artTime));
    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(bas
eStationNumber));
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(en
dTime));
    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(rangeRowQueryCri
teria));
        for (Row row : getRangeResponse.getRows()) {
            PrimaryKey curIndexPrimaryKey = row.getPrimaryKey();
            PrimaryKeyColumn mainCalledNumber = curIndexPrimaryKey.getPrimaryKeyColumn(PRIMARY_KE
Y_NAME_1);
            PrimaryKeyColumn callStartTime = curIndexPrimaryKey.getPrimaryKeyColumn(PRIMARY_KEY_NA
ME_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(ba
seStationNumber));
    startPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(st
artTime));
    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(bas
eStationNumber));
    endPrimaryKeyBuilder.addPrimaryKeyColumn(PRIMARY_KEY_NAME_2, PrimaryKeyValue.fromLong(en
dTime));
    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(rangeRowQueryCri
teria));
        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.

# 11.ApsaraDB RDS

# 11.1. Product Introduction

## 11.1.1. Features

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

### ApsaraDB RDS with different database engines

- ApsaraDB RDS for MySQL is developed based on AliSQL and provides excellent performance. ApsaraDB RDS for MySQL is a tried and tested solution that handled the high-volume concurrent traffic during Double 11. ApsaraDB RDS for MySQL supports deployment with mixed x86 and ARM clusters. It integrates 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 reading capabilities and increase the application throughput.
  - **Read/write splitting:** The read/write splitting feature provides a read/write splitting endpoint. This endpoint enables automatic read/write splitting for a primary instance and all of its read-only instances. An application can connect to the read/write splitting endpoint to read and write data. Write requests are routed to the primary instance while read requests are routed to read-only instances based on their weights. To scale up reading capabilities of the system, you need only to add more read-only instances.
- 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 provides a series of features including high availability, backup, and restoration to ease operations and maintenance loads.
- 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. ApsaraDB RDS for SQL Server provides basic features such as whitelist configuration, backup and restoration, transparent data encryption, data migration, and management for instances, accounts, and databases.
- PolarDB is a stable, secure, and scalable enterprise-class relational database service. PolarDB is developed based on PostgreSQL and offers enhanced 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.

### Scheduling service

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

## Data link service

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

- 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 updated, 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.

- 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 module 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.

## Specification change

When you change the specifications of an ApsaraDB RDS instance, the system automatically performs the following operations in the background:

1. Apply for resources required for the new instance.
2. Synchronize full data and incremental data from the original instance to the new instance.
3. When data synchronization is close to completion, set the original instance to the read-only state until all data is synchronized. After data synchronization is complete, disassociate the proxy IP address from the original instance and associates the proxy IP address with the new instance in the SLB backend.
4. Release the original instance and change the state of the new instance to running.

## Backup and restoration service

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

- 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 runs 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. ApsaraDB RDS supports the following backup methods:
  - Physical backup: directly backs up all files in all databases.
  - Logical backup: extracts data from the databases by using SQL statements and backs up the data in the text format.
- The Restoration module restores backup files from OSS to a destination node by using the following features:
  - Primary node rollback: rolls back the primary node to a specific point in time when an operation error occurs.
  - Secondary node repair: creates another 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.
- 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.

## Monitoring service

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

- The Service module tracks the status of services. For example, the Service module monitors whether other cloud services on which ApsaraDB RDS depends are running normally, such as SLB and 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 running normally.
- The Network module tracks the network status. The module monitors the connectivity between Elastic Compute Service (ECS) and ApsaraDB RDS and between physical servers of ApsaraDB RDS. It also monitors the packet loss rates on vRouters and vSwitches.
- The OS module tracks the statuses of hardware and OS kernel.
  - Hardware maintenance: The OS module constantly checks the operating status of the CPU, memory, motherboard, and storage device. It can predict faults and 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.
- The Instance module collects the following information about ApsaraDB RDS instances:
  - Availability information
  - Capacity and performance metrics

- SQL execution records

## High-availability service

The high-availability (HA) service uses the Detection, Repair, and Notice modules and a variety of HA polices to ensure the availability of data link services. It also processes internal database exceptions.

- The Detection module checks whether the primary and secondary nodes of the DB Engine are providing services normally. An 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 secondary nodes and heartbeat information from other HA nodes, provides a reference for the Detection module. All this information helps the module prevent improper judgment caused by exceptions such as network jitter. Failover can be completed within a short time.
- 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.
  - 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.
- The Notice module informs the SLB or Proxy module of status changes to the primary and secondary nodes to ensure that you always access the correct node.

  For example, assume that the Detection module discovers issues with the primary node and instructs the Repair module to resolve these issues. If the Repair module fails to resolve an issue, 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 another 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.
- Each HA policy defines a combination of service priorities and data replication modes to meet your business requirements.

  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:

- Synchronous
  - After an update that is initiated by your application is complete on the primary instance, the related log is synchronized to all the secondary instances. The transaction that is used to perform the update is considered committed only after at least one of the secondary instances receives and stores the log.
  - In synchronous mode, data replication remains synchronous and cannot be demoted to the asynchronous mode.
  - Synchronous replication is supported only when your database system consists of three or more instances. This means that only Enterprise Edition supports the synchronous mode. In addition, if the primary instance runs the Enterprise Edition, you cannot change the data replication mode.
- Semi-synchronous

  After an update that is initiated by your application is complete on the primary instance, the related log is synchronized to all the secondary instances. The transaction that is used to perform the update is considered committed after the secondary instances receive the log. Your database system does not need to wait for the secondary instances to replay the log.

  If the secondary instances are unavailable or a network interruption occurs between the primary and secondary instances, the semi-synchronous mode is demoted to the asynchronous mode.
- Asynchronous

  When your application initiates a request to add, delete, or modify data, the primary instance responds to your application immediately after it completes the requested operation. At the same time, the primary instance asynchronously replicates data to the secondary instances. In asynchronous mode, the unavailability of the secondary instances does not interrupt the workloads on the primary instance. However, the unavailability of the primary instance may cause data inconsistencies between the primary and secondary instances.

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

## Data migration service

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

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

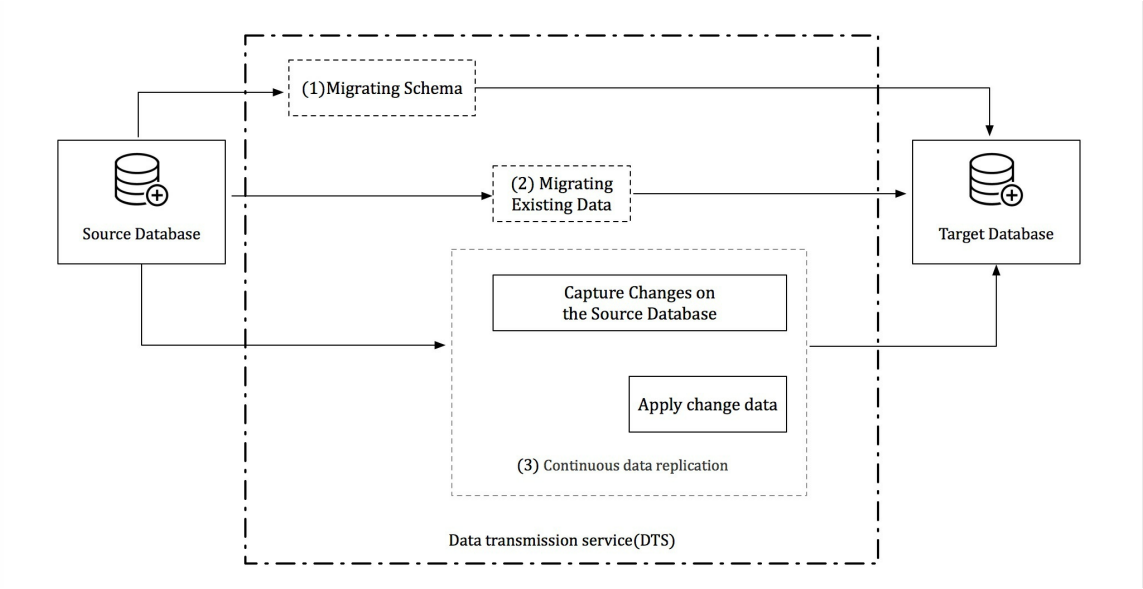
  DTS migrates all data of migration objects from the source database to the destination instance.

> **Important** 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.

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




## Dedicated instance family

The dedicated instance family of ApsaraDB RDS is more stable than other instance families because computing power, storage, and I/O resources are reserved for exclusive use by dedicated instances. In the dedicated instance family, dedicated host instance types have the highest specifications. A dedicated host instance occupies all resources of the physical host on which the instance is deployed.

- Resource isolation: To maintain stable computing performance, ApsaraDB RDS isolates computing resources among different instances. Dedicated instances are assigned exclusive CPU threads. This ensures that dedicated instances are not affected by the behavior of other instances on the physical host.
- Reserved storage: Dedicated instances have reserved storage and can provide more stable performance than general-purpose instances. If the disks on a host fail, dedicated instances can use the dual-node hot standby feature to switch over workloads to the other host without affecting user operations.

## Read/write splitting

- Easier maintenance by using a unified endpoint

If you do not enable the read/write splitting feature, you must add the endpoints of the primary and read-only instances to your application. This way, your database system routes write requests to the primary instance and read requests to the read-only instances.

If you enable the read/write splitting feature, you can use a dedicated proxy endpoint to implement read/write splitting. After your application is connected to this endpoint, your database system routes read and write requests to the primary and read-only instances based on the read weights of these instances. This reduces maintenance costs.

You can also create read-only instances to improve the read capability of your database system. You do not need to modify the configuration data on your application.

- Higher performance and lower maintenance costs by using a native link

  You can build your own proxy layer on the cloud to implement read/write splitting. In this case, data must be parsed and forwarded by multiple components before the data reaches your database system. As a result, response latencies increase. The read/write splitting feature is embedded in the ApsaraDB RDS ecosystem to reduce response latencies, increase processing speeds, and reduce maintenance costs.

- Ideal in a variety of use scenarios based on configurable read weights and thresholds

  You can specify the read weights of the primary and read-only instances. You can also specify the latency threshold for data replication to the read-only instances.

- High availability based on instance-level health checks

  The read/write splitting feature automatically checks the health status of the primary and read-only instances. If a read-only instance unexpectedly breaks down or its data replication latency exceeds the specified threshold, ApsaraDB RDS stops routing read requests to the instance. ApsaraDB RDS redirects read requests that are destined for the faulty read-only instance to healthy instances in your database system. This ensures service availability in the event of faults on individual read-only instances. After the faulty read-only instance is recovered, ApsaraDB RDS resumes routing read requests to the instance.

  > **Note** To avoid single points of failure (SPOFs), we recommend that you create at least two read-only instances.

## Data security

ApsaraDB RDS provides multiple network security features such as Virtual Private Cloud (VPC) and whitelists to ensure data security.

- VPC provides network isolation protection at the TCP layer.
- Anti-DDoS is supported to monitor and guard against Distributed-Denial-of-Service (DDoS) attacks.
- More than 1,000 IP addresses can be added to whitelists to prevent unauthorized access.
- Password authentication ensures secure and reliable access.

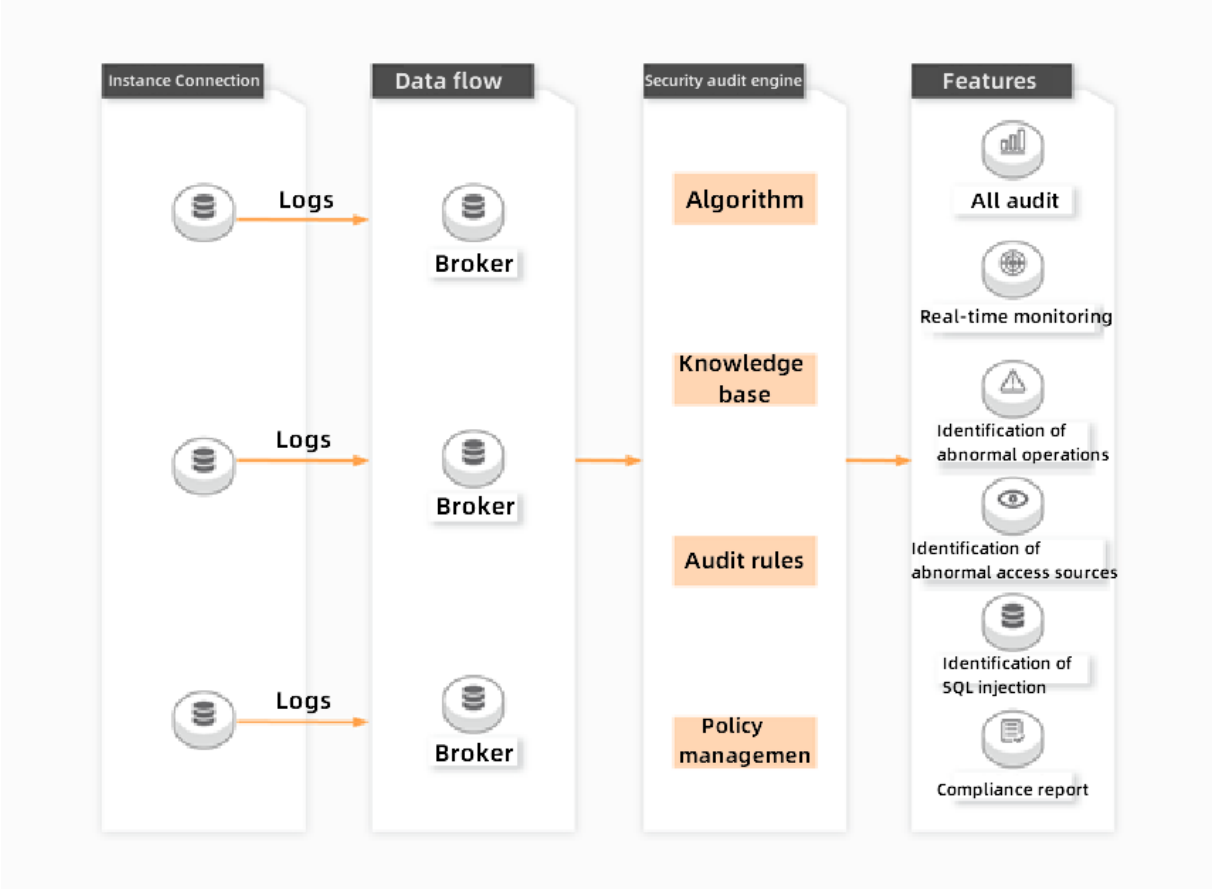
## SQL optimization technology

Automatic SQL optimization is a core feature provided by Database Autonomy Service (DAS). It uses autonomous capabilities to implement closed-loop SQL optimization.

- This feature detects workload exceptions, identifies database business changes, and identifies and locates problematic SQL statements. These statements include new slow SQL statements, SQL statements whose performance deteriorates, and inefficient SQL statements.
- For problematic SQL statements, this feature automatically uses the SQL diagnostics and optimization feature to generate optimization suggestions, such as creating optimal indexes, rewriting SQL statements, and recommending engines.
- This feature automatically assesses risks for optimization suggestions. It automatically generates canary release plans and orchestrates optimization tasks based on instance loads and profiles.
- This feature automatically selects maintenance windows and completes updates based on canary release plans. Indexes can be automatically released and updated.
- This feature tracks the optimization effect of updates from multiple dimensions and comprehensively assesses performance regression risks in real time. If the assessment result meets the expectation, optimization benefits are automatically calculated. Otherwise, the updates are automatically rolled back.

## SQL Audit

SQL Audit performs comprehensive and accurate audit on all database operations that are recorded. This feature provides real-time alerts for databases risks and generates compliance reports to improve the security of data assets.




SQL Audit can record all DML and DDL operations and provides a built-in security engine to audit database operations in real time.

- Detects attacks and threats in real time to identify security risks.
- Identifies high-risk operations by using a variety of intelligent algorithms and models.
- Automatically identifies new or abnormal access sources.

# 11.1.2. Benefits

## 11.1.2.1. 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.

## 11.1.2.2. 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.

## 11.1.2.3. 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.

## 11.1.2.4. 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.

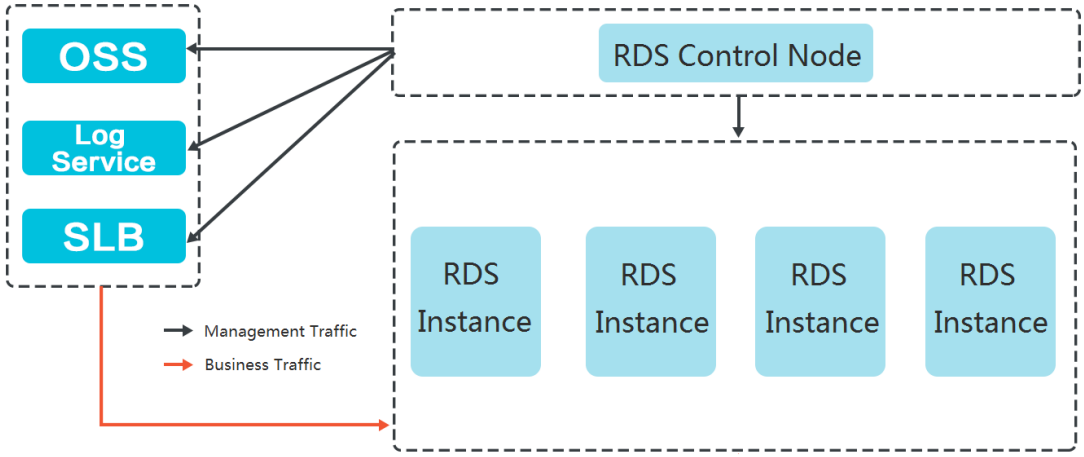
### Data restoration

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.

## 11.1.3. Architecture

The following figure shows the system architecture of ApsaraDB RDS.

ApsaraDB RDS system architecture




## 11.1.4. Features

### 11.1.4.1. Scheduling service

This topic describes the resource scheduling service of ApsaraDB RDS.

The scheduling service allocates and integrates underlying ApsaraDB RDS resources when you enable and migrate instances. When you use the console or call an API operation to create an ApsaraDB RDS instance, the scheduling service calculates the most suitable host to carry traffic to and from the instance. The scheduling service also allocates and integrates the underlying resources required to migrate ApsaraDB RDS instances. As instances are created, deleted, and migrated, the scheduling service calculates the fragmentation of the resources in zones, and then periodically integrates the resource fragments for these zones to handle more traffic data.

### 11.1.4.2. Data link service

ApsaraDB RDS provides data link services, such as 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 module or database engine runs on the server whose IP address is 192.168.0.1. The SLB module typically redirects all traffic destined for 10.1.1.1 to 192.168.0.1. If the server whose IP address is 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.

## 11.1.4.3. Instance specification change

This topic describes the operations performed in the background when you change the specifications of an ApsaraDB RDS instance.

When you change the specifications of an ApsaraDB RDS instance, the system performs the following operations in the background:

1. Apply for resources required by the new instance.
2. Perform full migration and incremental migration to synchronize data of the original instance to the new instance.
3. Change the IP address of the new instance. When data synchronization is close to completion, the original instance is set to the read-only state until all data is synchronized. After data synchronization is complete, the system disassociates the proxy IP address from the original instance and associates the proxy IP address with the new instance in the Server Load Balancer (SLB) backend.
4. Release the original instance and change the state of the new instance to running.

## 11.1.4.4. Backup and restoration service

This service supports data backup, restoration, and storage 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.

ApsaraDB RDS supports the following backup methods.

- Physical backup: directly backs up all files in all databases.
- Logical backup: extracts data from the databases by using SQL statements and backs up the data in the text format.

### Restoration

The Restoration 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 another 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.

## 11.1.4.5. 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

## 11.1.4.6. 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.

Examples:

- 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:

- Synchronous mode
  - After an update operation originated from an application is complete on a primary RDS instance, the update log record is synchronized to all secondary RDS instances that are attached to the primary RDS instance. The update transaction is considered committed after at least one of the secondary RDS instances receives and stores the update log record.
  - The synchronous mode cannot degrade to the asynchronous mode.
  - The synchronous mode is supported only when your database system consists of three or more RDS instances. This means that only RDS Enterprise Edition supports the synchronous mode. In addition, if you are using RDS Enterprise Edition, the data replication mode cannot be changed.
- Semi-synchronous mode

  After an update that is initialized by your application is complete on a primary RDS instance, the log is synchronized to all the secondary RDS instances. After the secondary RDS instances receive the log, the update transaction is considered committed. Your database system does not need to wait for the log to be replayed.

  If a secondary RDS instance is unavailable or the communication between a primary RDS instance and a secondary RDS instance is abnormal, the semi-synchronous mode degrades to the asynchronous mode.
- Asynchronous mode

After an add, delete, or modify operation originated from an application is complete on a primary RDS instance, the primary RDS instance immediately responds to the application. At the same time, the primary RDS instance asynchronously replicates the added, deleted, or modified data to the secondary RDS instances that are attached to the primary RDS instance. In asynchronous mode, the workloads on the primary RDS instance run as expected even if the secondary RDS instances are unavailable. However, if the primary RDS instance is unavailable, errors may occur due to data inconsistencies between the primary RDS instance and the secondary RDS instances.

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

# 11.1.4.7. 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

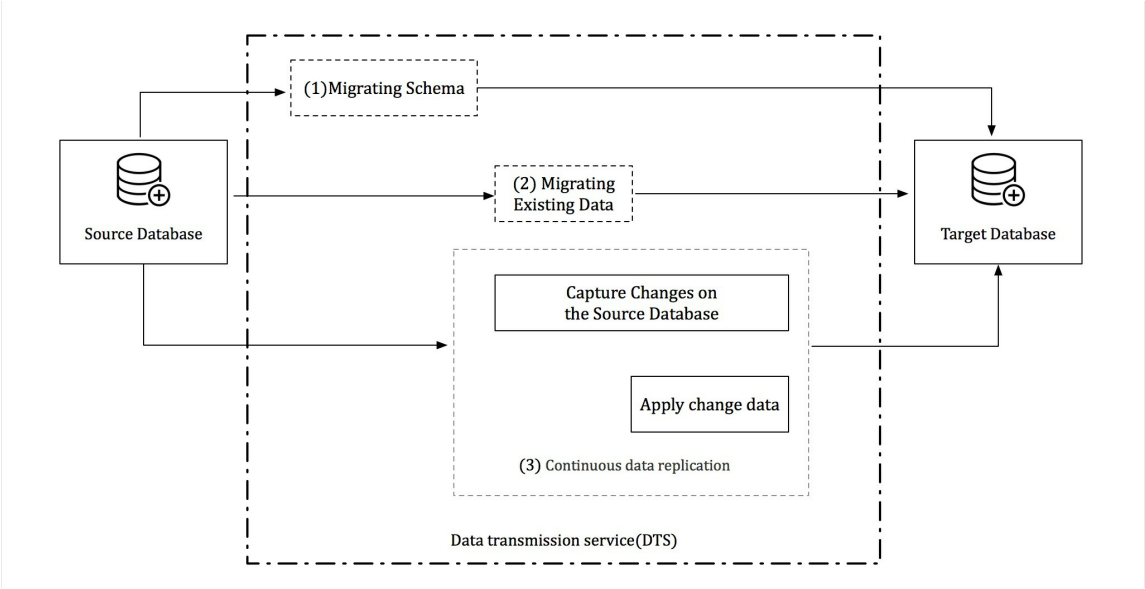
  DTS migrates all data of migration objects from the source database to the destination instance.

  > **Important** 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.

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




# 11.1.4.8. Dedicated instance family

This topic describes the dedicated instance family of ApsaraDB RDS.

## What is the ApsaraDB RDS dedicated instance family?

ApsaraDB RDS instances of the dedicated instance family have a fixed set of computing and storage resources and deliver stable I/O performance. The dedicated host instance type is the highest one among all dedicated instance types. An instance of this instance type uses all the resources of its physical server. For more information, see **Product Introduction > Instance types.**

## Features

- Isolated resources: To ensure the stability of computing performance, ApsaraDB RDS isolates the computing resources of dedicated instances. A dedicated instance exclusively occupies the allocated CPU threads and cores, so that its performance is not affected by the other instances on the physical server.
- Reserved storage space: The storage space of a dedicated instance is reserved specially for that instance. Therefore, dedicated instances deliver higher stability than general-purpose instances. The hot standby architecture allows failover to be automatically performed when a fault occurs on the disks of a specific server. This way, your business is not disrupted.

## Dedicated instance types

For more information, see **Product Introduction > Instance types.**

# 11.1.4.9. Read/write splitting

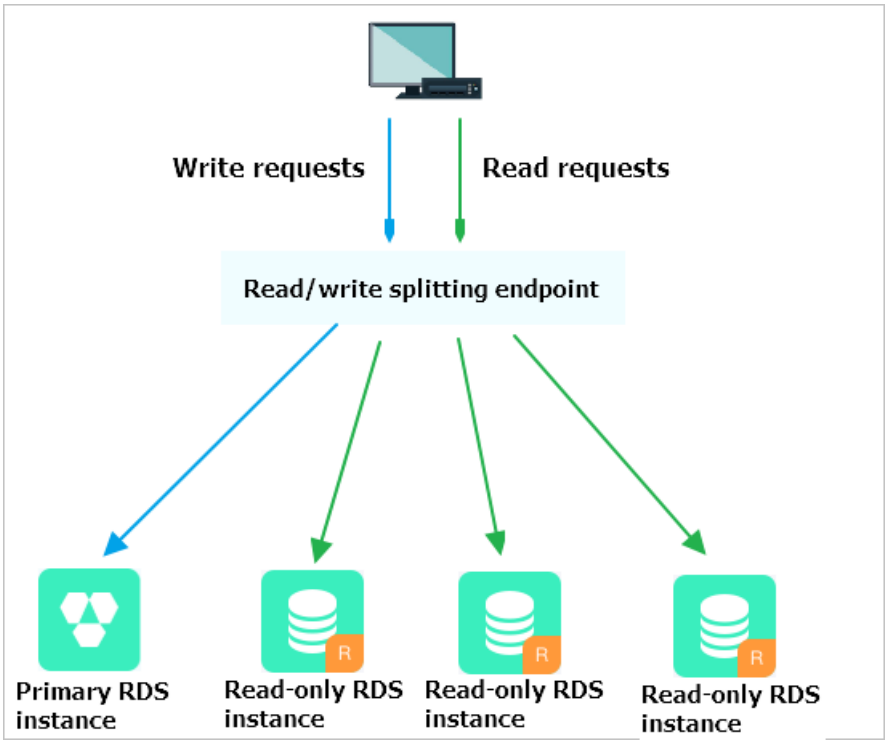
This topic describes how to use a dedicated proxy endpoint of an ApsaraDB RDS instance to implement read/write splitting. You must set the Read/Write Attribute parameter to Read/Write for the proxy terminal under which the used dedicated proxy endpoint is created.

## Background information

If your database system processes a large number of read requests and a small number of write requests, a single primary ApsaraDB RDS instance may fail to efficiently process the read requests. This may interrupt your workloads. In this case, you can create one or more read-only ApsaraDB RDS instances to offload read requests from the primary instance and increase the read capability of your database system.

After read-only instances are created, you can enable the read/write splitting feature. Then, you can use a dedicated proxy endpoint to implement read/write splitting. After your application is connected to this endpoint, ApsaraDB RDS routes write requests to the primary instance and read requests to the read-only instances based on the read weights of these instances.

If the internal or public endpoint of the primary ApsaraDB RDS instance is added to your application, all requests are routed to the primary ApsaraDB RDS instance. If you want to implement read/write splitting, you must add the endpoints and read weights of the primary and read-only ApsaraDB RDS instances to your application.




## Benefits

- Easier maintenance by using a unified endpoint

  If you do not enable the read/write splitting feature, you must add the endpoints of the primary and read-only instances to your application. After you add the endpoints, your database system routes write requests to the primary instance and read requests to the read-only instances.

  If you enable the read/write splitting feature, you can use a dedicated proxy endpoint to implement read/write splitting. After your application is connected to this endpoint, your database system routes read and write requests to the primary and read-only instances based on the read weights of these instances. This reduces maintenance costs.

  You can also improve the read capability of your database system by creating read-only instances. You do not need to modify the configuration data on your application.

- Higher performance and lower maintenance costs by using a native link

  You can build your own proxy layer on the cloud to implement read/write splitting. In this case, data needs to be parsed and forwarded by multiple components before the data reaches your database system. As a result, response latencies increase. The read/write splitting feature is built in the ApsaraDB RDS ecosystem and can efficiently reduce response latencies, increase processing speeds, and reduce maintenance costs.

- Ideal in various use scenarios based on configurable read weights and thresholds

  You can specify the read weights of the primary and read-only instances. You can also specify the latency threshold for data replication to the read-only instances.

- High availability based on instance-level health checks

  The read/write splitting feature enables ApsaraDB RDS to actively check the health status of the primary and read-only instances. If a read-only instance unexpectedly exits or its data replication latency exceeds the specified threshold, ApsaraDB RDS stops routing read requests to the instance. ApsaraDB RDS redirects the read requests that are destined for the faulty read-only instance to other healthy instances in your database system. This ensures service availability in the event of faults on individual read-only instances. After the faulty read-only instance is recovered, ApsaraDB RDS resumes routing read requests to the instance.

  > **Note** We recommend that you create at least two read-only ApsaraDB RDS instances to mitigate the impacts of single points of failure (SPOFs).

## Logic used to route requests

- The following requests are routed only to the primary instance:
  - Requests that are used to execute INSERT, UPDATE, DELETE, and SELECT FOR UPDATE statements
  - All requests that are used to perform data definition language (DDL) operations, such as the DDL operations to create databases or tables, delete databases or tables, and change schemas or permissions
  - All requests that are encapsulated in transactions
  - Requests for user-defined functions
  - Requests for stored procedures
  - Requests for EXECUTE statements
  - Requests for multi-statements
  - Requests that involve temporary tables
  - Requests for SELECT last_insert_id() statements
  - All requests to query or modify user environment variables
  - All requests for KILL statements in SQL (not KILL commands in Linux)
- The following requests are routed to the primary instance or its read-only instances:
  - Requests that are used to execute SELECT statements that are not encapsulated in transactions
  - Requests for COM_STMT_EXECUTE statements

- The following requests are routed to all the primary and read-only instances:
  - All requests to modify system environment variables
  - Requests for USE statements
  - Requests for SHOW PROCESSLIST statements

    > **Note** After a SHOW PROCESSLIST statement is executed, the dedicated proxy returns all the processes that run on the primary and read-only ApsaraDB RDS instances in your database system.

  - Requests for COM_STMT_PREPARE statements
  - Requests for COM_CHANGE_USER, COM_QUIT, and COM_SET_OPTION statements

# 11.1.4.10. Data security

ApsaraDB RDS provides various network security features, such as virtual private clouds (VPCs) and whitelists, to ensure data security.

ApsaraDB RDS provides the following network security features:

- Supports VPCs to isolate network environments at the TCP layer.
- Supports anti-DDoS to monitor and guard against Distributed-Denial-of-Service (DDoS) attacks.
- Allows you to configure more than 1,000 IP address whitelists to block malicious IP addresses.
- Supports password authentication to ensure secure and reliable access.

# 11.1.4.11. SQL optimization technology

## Background information

SQL optimization is a common practice among database administrators (DBAs) and application developers. SQL statements executed on databases are diversified. They continuously change in a dynamic way in scenarios such as rapid business iteration, changes of data distribution characteristics, hot spot changes, and database version upgrades. This makes SQL optimization indispensable.

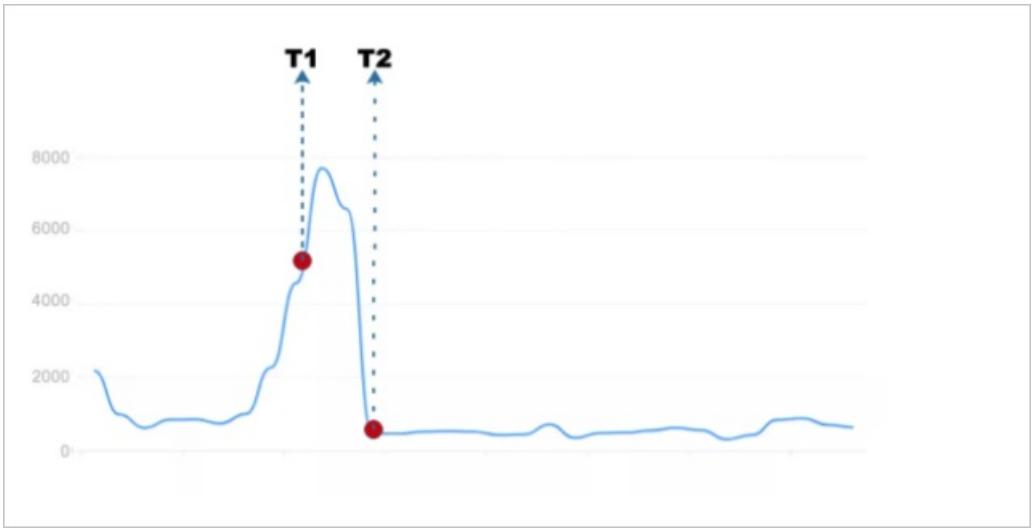
## Challenges

- How can I use comprehensive methods to perform troubleshooting in a quick and accurate manner? Slow query logs are not enough to analyze problematic SQL statements.
- How can database expertise or tools be used to accurately identify bottlenecks and obtain repair or optimization suggestions?
- How can pre-release security be assessed to comprehensively assess the optimization effects and impact (including side effects, such as the impact on related SQL statements and write operations)?
- How can I choose the phased release policy and the change window to promote online changes in a secure and stable way for complex deployment (such as large-scale sharding scenarios)?
- How can I continuously track the optimization effects to ensure optimization success?

## Issue risks

Two important time points are considered. The following figure shows a simple trend of a slow SQL statement. T1 represents the time point when the performance exception of the database instance is detected and the slow SQL statement starts to be optimized. T2 represents the time point when the optimization process is completed and the instance recovers to the normal state. During traditional optimization, this process entirely depends on manual operations. This has the following two serious weaknesses:

- T1 is much later than the expected time. This indicates that the exception is not detected or responded in a timely manner. Even if the exception is detected, it may have existed for a long time and have been on the edge of failure.
- If the value of T2-T1 represents a long processing time, user experience is seriously affected and the failure risk is greatly increased.




In addition to the preceding two issues, you may have also faced the following challenges:

- How can I achieve continuous optimization? Detect issues and optimize SQL statements in a timely manner to prevent issues from being accumulated. This ensures both the stability and the continuous optimal running status of database instances.
- How can I shorten the processing duration, minimize the impact, and use the comprehensive method to ensure the stability of database instances and solve symptoms and root causes?

The traditional method is human-driven. This makes the two limits obvious. This method is often fault-driven and is incapable of coping with a large number of issues. As the business scale and the instance scale grow, all these issues are magnified. Even if manpower is increased, the issues cannot be resolved at a high probability. This forms a vicious cycle.

## Methods

Automatic SQL optimization is a core service of Alibaba Cloud Database Autonomy Service (DAS). It provides the self-optimization feature, which is the autonomous capability to achieve the closed loop of SQL optimization.

The closed loop capability is achieved in the following aspects:

- The capability detects workload exceptions, identifies database business changes, and identifies and locates problematic SQL statements. The statements include new slow SQL

statements, SQL statements whose performance deteriorates, and inefficient SQL statements.

- For problematic SQL statements, the capability automatically invokes the SQL diagnostics and optimization service to generate optimization suggestions, such as creating optimal indexes, rewriting SQL statements, and recommending engines.
- The capability automatically completes risk assessment of optimization suggestions. It automatically generates a phased plan and automatically orchestrates optimization tasks by using the load status and the profile of the database instance.
- The capability automatically selects the maintenance window, and completes relevant online changes by using the phased plan. In the current phase, indexes can be automatically released and changed.
- The capability starts multi-dimensional optimization effect tracking for released changes to continuously and comprehensively assess performance regression risks in real time. If the assessment result is expected, optimization benefits are automatically calculated. If the assessment result is not expected, the changes are automatically rolled back.

Manpower-intensive passive optimization is transformed into intelligence-based active and continuous optimization based on the closed loop of automatic SQL optimization. This achieves unattended SQL optimization in the end. The closed loop of automatic SQL optimization works like a group of database experts who provide the 24/7 guarding service to take care of your databases. In addition, they keep your database system running in the optimal optimization state.

- In the process of achieving the preceding goal, the following challenges are faced:
  - Accuracy: An exception detection mechanism must be constructed to accurately identify the optimization time and accurately locate problematic SQL statements.
  - Professional diagnostics: Powerful professional optimization and diagnostics must be available to support the goal. If valid professional diagnostics is unavailable, SQL optimization cannot be implemented.
  - Security: Everything online is important. Online changes must be secure and controllable.
  - Comprehensiveness: Comprehensive multi-dimensional tracking and comprehensive real-time assessment of optimization effects are also required to ensure security.
  - Linkage: Sometimes, complex online issues, such as malicious slow SQL statements that abruptly occur, need to be comprehensively resolved. Therefore, automatic SQL throttling and automatic SQL optimization of DAS must be linked to address both symptoms and root causes of the issues.
  - Scale: A service architecture that provides sufficient scalability must be built to support automatic optimization for hundreds of thousands of and millions of servers.

## Methods

1. Implementation architecture

   Automatic SQL optimization of DAS is a data-driven closed loop.

   - Exception events: Exception events are the fuse for triggering automatic SQL optimization. The DAS event center allows you to perform centralized management on exception events. Exception events are generated in scenarios or the system, such as real-time exception detection, offline analysis, and workload detection, and the alert system.
   - Diagnostics initialization: After the automatic SQL optimization service receives an exception

event from the event center, it performs preliminary diagnostics for the instance and initiates a diagnostic request to the diagnostic engine. Then, the service processes the diagnosis result (one or more suggestions). After the result is processed, this service completes effectiveness assessment, generates a new optimization event, and sends the event to the event center to drive the subsequent optimization process.

- Suggestion push: After a user enters the DAS autonomy center, the user can choose whether to accept optimization suggestions when the autonomy service is disabled. The subsequent automatic optimization process can be triggered based on the self-decision result.
- Change release: Select the maintenance window to issue change commands and determine the command running status.
- Effect tracking and measurement: When the optimization suggestions take effect, the decision engine starts a tracking task to track the performance of optimized SQL statements and related SQL statements. If the performance deteriorates, the SQL statements are automatically rolled back. In general, if no rollback occurs after the performance is tracked for 24 hours, benefits are calculated.

## Issue detection

SQL optimization allows you to detect SQL exceptions in the following three scenarios:

- SQL optimization is regularly triggered. In a regular maintenance window, slow SQL statements executed on user instances are regularly analyzed offline to initialize SQL optimization.
- SQL optimization is triggered when the performance of some SQL statements deteriorates. When the workload exception detection algorithm detects SQL statements whose performance deteriorates in real time, automatic SQL optimization is triggered. For complex online problems, automatic SQL optimization and automatic SQL throttling of DAS need to be linked to trigger automatic SQL optimization.
- SQL optimization is triggered when the instance workload changes. When business SQL statements are published or unpublished, the database load and the data volume change, and the existing indexes cannot meet the performance requirements of the current business. Therefore, diagnostics and optimization at the instance workload level are triggered.

## Diagnostics capability

The SQL diagnostics and optimization service of DAS provides powerful support for automatic SQL optimization. This service considers optimization issues by using the cost-based model in the same way as the database optimizer. In the end, this service implements quantitative assessment on all the possible recommendation options based on the execution cost and make reliable recommendations.

This service has stably run in Alibaba Group for nearly three years. It can diagnose about 50,000 SQL statements, and supports SQL optimization for business applications in the entire group. Over the past three years, the SQL diagnostics success rate has remained more than 98%, and the recommendation rate for slow SQL statements has remained more than 75%.

## Security changes

Security change includes security check before change, the phased change policy, and performance tracking after change.

- Security check: To reduce risks, changes occur in only the maintenance window. In addition,

changes occur only when each metric value of the replication delay between primary and secondary instances, instance load, and tablespace is within the secure range.

- Phased change policy: For example, when sharding involves a large amount of data, a phased plan is automatically generated to implement changes in batches. This reduces risks. During the change process, the system monitors the replication delay between primary and secondary instances. When the latency exceeds the threshold, the system immediately suspends each index change task of the database, and makes sure that only one change task is run for each database.
- Effect evaluation: The effect evaluation algorithm tracks performance of optimized SQL statements and related SQL templates to prevent failures that are caused by the deteriorated performance. The performance tracking algorithm compares performance metrics of an SQL template before the optimization with those after the optimization by using the decision tree model. This algorithm comprehensively determines whether the SQL template performance deteriorates at the time. Service changes occur on a daily basis. The default tracking time is 24 hours. If no rollback occurs, the optimization is successful and the actual optimization benefits are calculated.

# 11.1.4.12. SQL audit

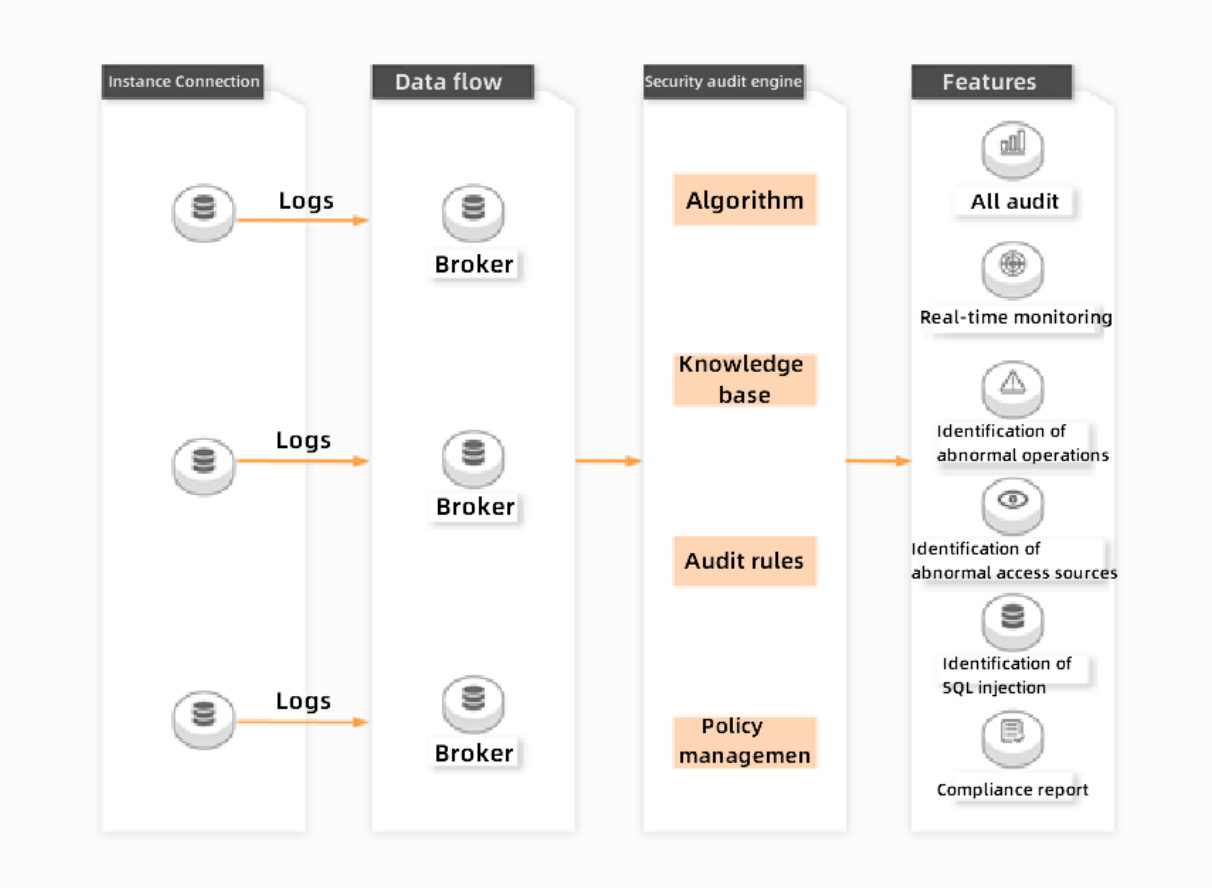
This topic describes the SQL audit feature of ApsaraDB RDS.

## Background information

Data is one of the most valuable assets of an enterprise but is prone to internal and external security issues such as data breaches, data corruption, and hacker attacks. Major and trivial events related to such issues frequently occur across the world. To ensure data security, an enterprise must have the capabilities to detect, identify, and guard against abnormal access to its databases.

SQL audit is a core feature to ensure data security. It records all operations in the database and performs comprehensive and accurate auditing on the operations. It also sends alerts against risks in real time and generates compliance reports.

## Architecture




When the SQL audit feature is enabled, all DML and DDL operations are recorded. The built-in security engine performs auditing on the operations in real time.

- Detects attacks and threats in real time to avoid security risks.
- Identifies high-risk operations based on algorithms and numerous models.
- Automatically identifies new or abnormal sources.

### Scenarios

- Your ApsaraDB RDS instance is used for sectors that require high data security. These sectors include finance, security, stocks, public service, and insurance.
- You need to analyze the running status of your ApsaraDB RDS instance to perform troubleshooting or to check the performance of SQL statements. Issues may occur in extreme circumstances.
- You need to restore the data of your ApsaraDB RDS instance by using the logged information of the executed SQL statements. This restoration is required in extreme circumstances.

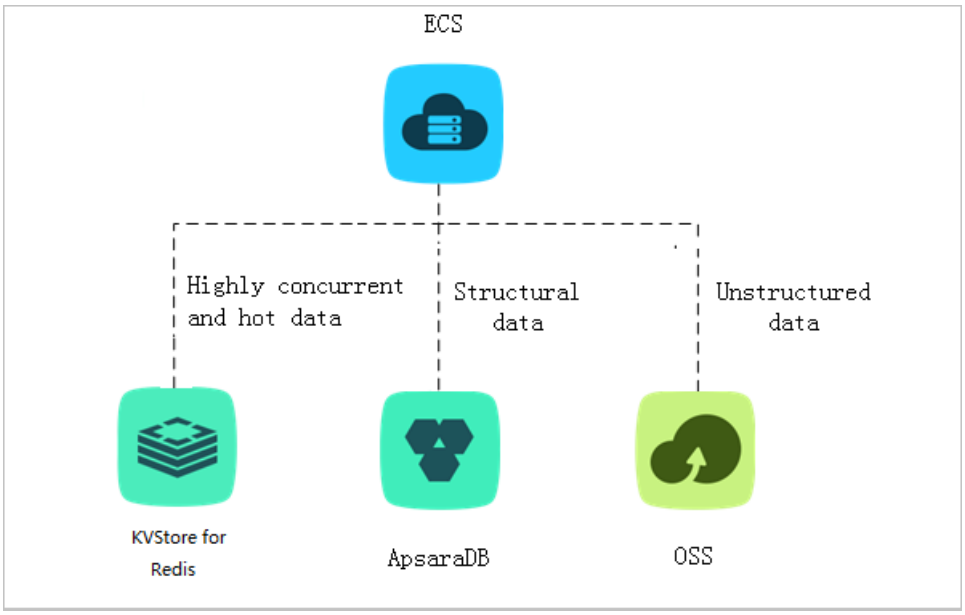
## 11.1.5. Scenarios

## 11.1.5.1. 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 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 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

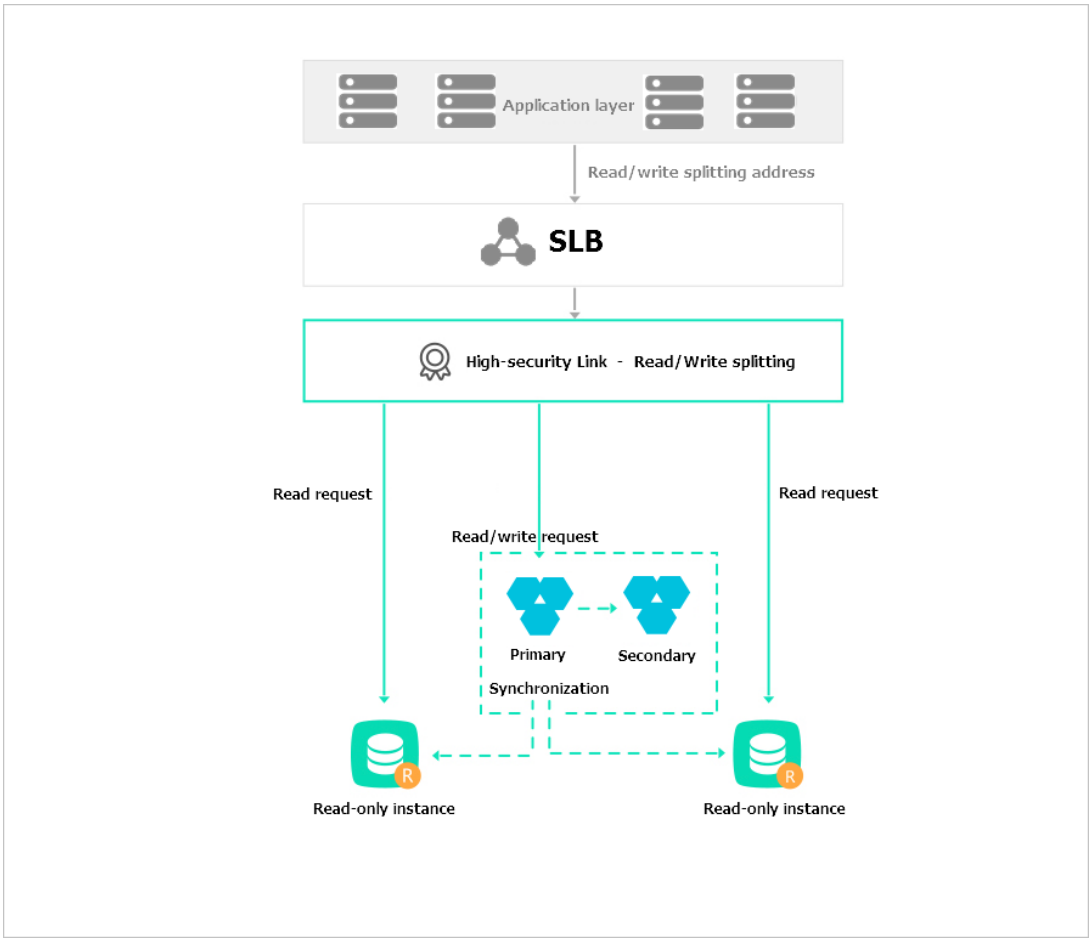
OSS is a secure, reliable, low-cost, and high-capacity storage service offered by Alibaba Cloud. ApsaraDB RDS can be used in conjunction with OSS to implement a multi-type data storage solution. For example, assume that ApsaraDB RDS and OSS are jointly used to set up an online forum. Resources such as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

# 11.1.5.2. 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

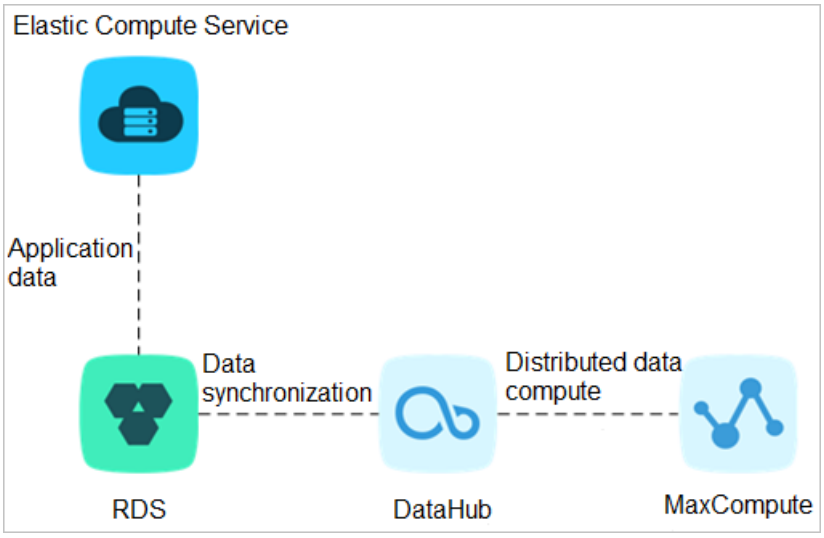



## 11.1.5.3. Big data analysis

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




# 11.1.6. Limits

## 11.1.6.1. 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).<br>• 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.<br>• 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.<br>• Data can be imported by using the MySQL CLI or DTS. |

| Operation | Limit |
| --- | --- |
| 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 MylSAM engine. MyISAM engine tables in new instances are converted to InnoDB engine tables.<br>• For performance and security considerations, we recommend that you use the InnoDB storage engine.<br>• 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. |
| 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.<br>• The ApsaraDB RDS console allows you to manage accounts and databases.<br>• Instances that support standard accounts also support privileged accounts. |
| Privileged account | • Custom authorization is supported.<br>• The ApsaraDB RDS console does not provide interfaces to manage accounts or databases. These operations can be performed only by using code or DMS.<br>• The privileged account cannot be reverted back to a standard account. |

# 11.1.6.2. 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 |

| Operation | Limit |
| --- | --- |
| 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 |
| 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 |

# 11.1.6.3. 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. |

| Operation | Limit |
| --- | --- |
| Instance restart | Instances must be restarted by using the ApsaraDB RDS console or API operations. |

## 11.1.6.4. 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. |
| Database replication | • The system automatically builds HA databases based on PolarDB streaming replication without user input.<br>• Secondary PolarDB instances are hidden and cannot be accessed. |
| Instance restart | Instances must be restarted by using the ApsaraDB RDS console or API operations. |

## 11.1.7. 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. |

| Term | Description |
| --- | --- |
| 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. |
| 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. |

# 11.1.8. Instance types

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

## IOPS

The maximum 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 × Storage capacity, 25000}
```

**Note** If the throughput of an RDS 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 specifications | Maximum number of connections | Storage | |
|---|---|---|---|---|---|---|
| | | | | | Maximum IOPS | Storage capacity |
| | General-purpose instance family | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.t1.small | 1 core, 1 GB | 300 | 600 | 5 GB to 2,000 GB |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.s1.small | 1 core, 2 GB | 600 | 1,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.s2.large | 2 cores, 4 GB | 1,200 | 2,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.s2.xlarge | 2 cores, 8 GB | 2,000 | 4,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.s3.large | 4 cores, 8 GB | 2,000 | 5,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.m1.medium | 4 cores, 16 GB | 4,000 | 7,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.c1.large | 8 cores, 16 GB | 4,000 | 8,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.c1.xlarge | 8 cores, 32 GB | 8,000 | 12,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.c2.xlarge | 16 cores, 64 GB | 16,000 | 14,000 | |

| RDS edition | Instance family | Instance type | CPU and memory specifications | Maximum number of connections | Storage | |
|---|---|---|---|---|---|---|
| | | | | | Maximum IOPS | Storage capacity 5 GB to 3,000 GB |
| High-availability Edition | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.c2.xlp2 | 16 cores, 96 GB | 24,000 | 16,000 | |
| | | MySQL 5.6 and MySQL 5.7<br>rds.mysql.c2.2xlarge | 16 cores, 128 GB | 32,000 | 16,000 | |
| | Dedicated instance family | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>mysql.x4.large.2 | 4 cores, 16 GB | 2,500 | 4,500 | 50 GB to 1,000 GB |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>mysql.x4.xlarge.2 | 8 cores, 32 GB | 5,000 | 9,000 | 500 GB to 3,000 GB |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>mysql.x4.2xlarge.2 | 16 cores, 64 GB | 10,000 | 18,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>mysql.x4.4xlarge.2 | 32 cores, 128 GB | 20,000 | 36,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>mysql.x8.medium.2 | 2 cores, 16 GB | 2,500 | 4,500 | 50 GB to 1,000 GB |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>mysql.x8.large.2 | 4 cores, 32 GB | 5,000 | 9,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>mysql.x8.xlarge.2 | 8 cores, 64 GB | 10,000 | 18,000 | |

| RDS edition | Instance family | Instance type | CPU and memory specifications | Maximum number of connections | Storage | 500 GB to 3,000 GB |
|---|---|---|---|---|---|---|
| | | | | | Maximum IOPS | Storage capacity |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>mysql.x8.2xlarge.2 | 16 cores, 128 GB | 20,000 | 36,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>mysql.x8.4xlarge.2 | 32 cores, 256 GB | 40,000 | 72,000 | 1,000 GB to 6,000 GB |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>mysql.x8.8xlarge.2 | 64 cores, 512 GB memory | 80,000 | 144,000 | |
| | Dedicated host instance family | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.st.v52 | 90 cores, 720 GB | 150,000 | 140,000 | 1,000 GB to 6,000 GB |
| | | MySQL 5.6 and MySQL 5.7<br>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 specifications | Maximum number of connections | Storage | |
|---|---|---|---|---|---|---|
| | | | | | Maximum IOPS | Storage capacity |
| | | MySQL 5.7<br>mysql.n2.small.25 | 1 core, 2 GB | 600 | 1,000 | 1,000 GB to 6,000 GB |
| | | MySQL 5.7<br>mysql.n2.medium.25 | 2 cores, 4 GB | 1,200 | 2,000 | |
| | | MySQL 5.7<br>mysql.n4.medium.25 | 2 cores, 8 GB | 2,000 | 4,000 | |

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

| RDS edition | Instance family | Instance type | CPU and memory specifications | Maximum number of connections | Storage | |
|---|---|---|---|---|---|---|
| | | | | | Maximum IOPS | Storage capacity |
| | Dedicated instance family (with a large memory capacity) | MySQL 5.7<br>mysql.x8.medium.25 | 2 cores, 16 GB | 2,500 | 4,500 | 50 GB to 1,000 GB |
| | | MySQL 5.7<br>mysql.x8.large.25 | 4 cores, 32 GB | 5,000 | 9,000 | |
| | | MySQL 5.7<br>mysql.x8.xlarge.25 | 8 cores, 64 GB | 10,000 | 18,000 | 500 GB to 3,000 GB |
| | | MySQL 5.7<br>mysql.x8.2xlarge.25 | 16 cores, 128 GB | 20,000 | 36,000 | |
| | | MySQL 5.7<br>mysql.x8.4xlarge.25 | 32 cores, 256 GB | 40,000 | 72,000 | 1,000 GB to 3,000 GB |
| | Dedicated host instance family | MySQL 5.7<br>mysql.st.8xlarge.25 | 60 cores, 470 GB | 100,000 | 120,000 | 1,000 GB to 6,000 GB |
| | | MySQL 5.7<br>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 specifications | Maximum number of connections | Storage | |
|---|---|---|---|---|---|---|
| | | | | | Maximum IOPS | Storage capacity |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.t1.small | 1 core, 1 GB | 300 | 600 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.s1.small | 1 core, 2 GB | 600 | 1,000 | |

| Instance role | Instance family | Instance type | CPU and memory specifications | Maximum number of connections | Storage | |
|---|---|---|---|---|---|---|
| | | | | | Maximum IOPS | Storage capacity |
| | General-purpose instance family | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.s2.large | 2 cores, 4 GB | 1,200 | 2,000 | 5 GB to 2,000 GB |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.s2.xlarge | 2 cores, 8 GB | 2,000 | 4,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.s3.large | 4 cores, 8 GB | 2,000 | 5,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.m1.medium | 4 cores, 16 GB | 4,000 | 7,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.c1.large | 8 cores, 16 GB | 4,000 | 8,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.c1.xlarge | 8 cores, 32 GB | 8,000 | 12,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.c2.xlarge | 16 cores, 64 GB | 16,000 | 14,000 | 5 GB to 3,000 GB |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.c2.xlp2 | 16 cores, 96 GB | 24,000 | 16,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>mysqlro.x4.large.1 | 4 cores, 16 GB | 2,500 | 4,500 | |

| Read-only instance role | Instance family | Instance type | CPU and memory specifications | Maximum number of connections | Storage: Maximum IOPS | Storage: 50 GB to 1,000 GB Storage capacity |
|---|---|---|---|---|---|---|
| | Dedicated instance family | MySQL 5.6, MySQL 5.7, and MySQL 8.0 mysqlro.x4.xlarge.1 | 8 cores, 32 GB | 5,000 | 9,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0 mysqlro.x4.2xlarge.1 | 16 cores, 64 GB | 10,000 | 18,000 | 500 GB to 3,000 GB |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0 mysqlro.x4.4xlarge.1 | 32 cores, 128 GB | 20,000 | 36,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0 mysqlro.x8.medium.1 | 2 cores, 16 GB | 2,500 | 4,500 | 50 GB to 1,000 GB |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0 mysqlro.x8.large.1 | 4 cores, 32 GB | 5,000 | 9,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0 mysqlro.x8.xlarge.1 | 8 cores, 64 GB | 10,000 | 18,000 | 500 GB to 3,000 GB |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0 mysqlro.x8.2xlarge.1 | 16 cores, 128 GB | 20,000 | 36,000 | |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0 mysqlro.x8.4xlarge.1 | 32 cores, 256 GB | 40,000 | 72,000 | 1,000 GB to 6,000 GB |
| | | MySQL 5.6, MySQL 5.7, and MySQL 8.0 mysqlro.x8.8xlarge.1 | 64 cores, 512 GB | 80,000 | 144,000 | |

| Instance role | Instance family | Instance type | CPU and memory specifications | Maximum number of connections | Storage | |
|---|---|---|---|---|---|---|
| | | | | | Maximum IOPS | Storage capacity |
| | Dedicated host instance family | MySQL 5.6, MySQL 5.7, and MySQL 8.0<br>rds.mysql.st.v52 | 90 cores, 720 GB | 150,000 | 140,000 | 1,000 GB to 6,000 GB |
| | | MySQL 5.6 and MySQL 5.7<br>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 specifications | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|
| | Dedicated instance family (with a large memory capacity) | PostgreSQL 9.4 and PostgreSQL 10.0<br>pg.x8.medium.2 | 2 cores, 16 GB | 2,500 | 4,500 | |
| | | 9.4 and 10.0<br>pg.x8.large.2 | 4 CPU cores, 32 GB memory | 5,000 | 9,000 | |
| | | PostgreSQL 9.4 and PostgreSQL 10.0<br>pg.x8.xlarge.2 | 8 cores, 64 GB | 10,000 | 18,000 | |
| | | PostgreSQL 9.4 and PostgreSQL 10.0<br>pg.x8.2xlarge.2 | 16 cores, 128 GB | 12,000 | 36,000 | |

| RDS edition | Instance family | Instance type | CPU and memory specifications | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|
| • High-availability Edition<br>• Read-only instance | Dedicated instance family (with a large number of cores) | PostgreSQL 9.4 and PostgreSQL 10.0<br>pg.x4.large.2 | 4 cores, 16 GB | 2,500 | 4,500 | 20 GB to 6,000 GB<br>The storage capacity increases in increments of 5 GB. |
| | | PostgreSQL 9.4 and PostgreSQL 10.0<br>pg.x4.xlarge.2 | 8 cores, 32 GB | 5,000 | 9,000 | |
| | | PostgreSQL 9.4 and PostgreSQL 10.0<br>pg.x4.2xlarge.2 | 16 cores, 64 GB | 10,000 | 18,000 | |
| | | PostgreSQL 9.4 and PostgreSQL 10.0<br>pg.x4.4xlarge.2 | 32 cores, 128 GB | 12,000 | 36,000 | |
| | Dedicated host instance family | PostgreSQL 9.4 and PostgreSQL 10.0<br>rds.pg.st.h43 | 60 cores, 470 GB | 12,000 | 50,000 | |

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

| RDS edition | Instance family | Instance type | CPU and memory specifications | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|
| | General-purpose instance family | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.n2.small.2c | 1 core, 2 GB | 200 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.n2.medium.2c | 2 cores, 4 GB | 100 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x4.medium.2c | 2 cores, 8 GB | 800 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x8.medium.2c | 2 cores, 16 GB | 1,600 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x2.large.2c | 4 cores, 8 GB | 800 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x4.large.2c | 4 cores, 16 GB | 1,600 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x8.large.2c | 4 cores, 32 GB | 3,200 | | |

| RDS edition | Instance family | Instance type | CPU and memory specifications | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x2.xlarge.2c | 8 cores, 16 GB | 1,600 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x4.xlarge.2c | 8 cores, 32 GB | 3,200 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x8.xlarge.2c | 8 cores, 64 GB | 6,400 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x2.3large.2c | 12 cores, 24 GB | 2,400 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x4.3large.2c | 12 cores, 48 GB | 4,800 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x8.3large.2c | 12 cores, 96 GB | 9,600 | | |

| RDS edition | Instance family | Instance type | CPU and memory specifications | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|
| High-availability Edition | Dedicated instance family | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x2.2xlarge.2c | 16 cores, 32 GB | 3,200 | IOPS | 20 GB to 6,000 GB |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x4.2xlarge.2c | 16 cores, 64 GB | 6,400 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x8.2xlarge.2c | 16 cores, 128 GB | 12,800 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x2.3xlarge2c | 24 cores, 48 GB | 4,800 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x4.3xlarge.2c | 24 cores, 96 GB | 9,600 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x8.3xlarge.2c | 24 cores, 192 GB | 19,200 | | |

| RDS edition | Instance family | Instance type | CPU and memory specifications | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x2.4xlarge.2c | 32 cores, 64 GB | 6,400 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x4.4xlarge.2c | 32 cores, 128 GB | 12,800 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x8.4xlarge.2c | 32 cores, 256 GB | 25,600 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x2.13large.2c | 52 cores, 104 GB | 10,400 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x4.13large.2c | 52 cores, 192 GB | 19,200 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x8.13large.2c | 52 cores, 384 GB | 38,400 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x2.8xlarge.2c | 64 cores, 128 GB | 12,800 | | |

| RDS edition | Instance family | Instance type | CPU and memory specifications | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x4.8xlarge.2c | 64 cores, 256 GB | 25,600 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x8.8xlarge.2c | 64 cores, 512 GB | 51,200 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x2.13xlarge.2c | 104 cores, 192 GB | 19,200 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>pg.x4.13xlarge.2c | 104 cores, 384 GB | 38,400 | | |
| | | PostgreSQL 10.0, PostgreSQL 11.0, PostgreSQL 12.0, and PostgreSQL 13.0<br>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 capacity | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|---|
| | 2008 R2 | Dedicated instance family | 2008r2<br>mssql.x8.medium.2 | 2 cores, 16 GB | 2,500 | 4,500 | 250 GB |
| | | | 2008r2<br>mssql.x8.large.2 | 4 cores, 32 GB | 5,000 | 9,000 | 500 GB |
| | | | 2008r2<br>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 instance family | 2008r2<br>rds.mssql.st.d13 | 30 cores, 220 GB | 64,000 | 20,000 | 2,000 GB |
| | | | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x4.medium.e2 | 2 cores, 8 GB | | | |
| | | | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x8.medium.e2 | 2 cores, 16 GB | | | |
| | | | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x4.large.e2 | 4 cores, 16 GB | | | |
| | | | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x8.large.e2 | 4 cores, 32 GB | | | |

| RDS edition | SQL Server version | Instance family | Instance type | CPU and memory capacity | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|---|
| High-availability Edition | 2012 EE: 2012_ent_ha<br>2016 EE: 2016_ent_ha<br>2017 EE | Dedicated instance family | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x4.xlarge.e2 | 8 cores, 32 GB | | | |
| | | | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x8.xlarge.e2 | 8 cores, 64 GB | | | |
| | | | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x4.2xlarge.e2 | 16 cores, 64 GB | | | |
| | | | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x8.2xlarge.e2 | 16 cores, 128 GB | | | |
| | | | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x4.3xlarge.e2 | 24 cores, 96 GB | | | |
| | | | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x4.4xlarge.e2 | 32 cores, 128 GB | | | |
| | | | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x8.4xlarge.e2 | 32 cores, 256 GB | | | |
| | | | | | Unlimi | For more infor | 20 GB to |

| RDS edition | SQL Server version | Instance family | Instance type | CPU and memory capacity | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|---|
| | | | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x8.7xlarge.e2 | 56 cores, 480 GB | Unlimited. For more information, see IOPS. | | 20 GB to 4,000 GB |
| | | | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x4.8xlarge.e2 | 64 cores, 256 GB | | | |
| | | | 2012_ent_ha, 2016_ent_ha, and 2017_ent<br>mssql.x8.8xlarge.e2 | 64 cores, 512 GB | | | |
| | | | 2012_std_ha and 2016_std_ha<br>mssql.x4.medium.s2 | 2 cores, 8 GB | | | |
| | | | 2012_std_ha and 2016_std_ha<br>mssql.x8.medium.s2 | 2 cores, 16 GB | | | |
| | | | 2012_std_ha and 2016_std_ha<br>mssql.x4.large.s2 | 4 cores, 16 GB | | | |
| | | | 2012_std_ha and 2016_std_ha<br>mssql.x8.large.s2 | 4 cores, 32 GB | | | |
| | | | 2012_std_ha and 2016_std_ha<br>mssql.x4.xlarge.s2 | 8 cores, 32 GB | | | |
| | | Dedicate | | | | | |

| RDS edition | SQL Server version | Instance family | Instance type | CPU and memory capacity | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|---|
| | 2012 SE: 2012_std_ha<br>2016 SE: 2016_std_ha | d instance family | 2012_std_ha and 2016_std_ha<br>mssql.x8.xlarge.s2 | 8 cores, 64 GB | | | |
| | | | 2012_std_ha and 2016_std_ha<br>mssql.x4.2xlarge.s2 | 16 cores, 64 GB | | | |
| | | | 2012_std_ha and 2016_std_ha<br>mssql.x8.2xlarge.s2 | 16 cores, 128 GB | | | |
| | | | 2012_std_ha and 2016_std_ha<br>mssql.x4.3xlarge.s2 | 24 cores, 96 GB | | | |

## PolarDB (High-availability Edition)

| RDS edition | Engine version | Instance family | Instance type | CPU and memory capacity | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|---|
| | | | 11<br>polardb.x4.small.2 | 1 core, 4 GB | 200 | 5,000 | 250 GB |
| | | | 11<br>polardb.x4.medium.2 | 2 cores, 8 GB | 400 | 10,000 | |
| | | | 11<br>polardb.x8.medium.2 | 2 cores, 16 GB | 2,500 | 15,000 | |

| RDS edition | Engine version | Instance family | Instance type | CPU and memory capacity | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|---|
| | 11 | Dedicated instance family | 11 polardb.x4.large.2 | 4 cores, 16 GB | 2,500 | 20,000 | 250 GB to 500 GB<br>The storage capacity increases in increments of 5 GB. |
| | | | 11 polardb.x8.large.2 | 4 cores, 32 GB | 5,000 | 30,000 | |
| | | | 11 polardb.x4.xlarge.2 | 8 cores, 32 GB | 5,000 | 40,000 | 500 GB to 1,000 GB<br>The storage capacity increases in increments of 5 GB. |
| | | | 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 GB to 2,000 GB<br>The storage capacity increases in increments of 5 GB. |
| | | | 11 polardb.x8.2xlarge.2 | 16 cores, 128 GB | 12,000 | 120,000 | |

| RDS edition | Engine version | Instance family | Instance type | CPU and memory capacity | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|---|
| • High-availability Edition<br>• Read-only instance | | | 11<br>polardb.x4.4xlarge.2 | 32 cores, 128 GB | 12,000 | 160,000 | 2,000GB to 3,000 GB<br>The storage capacity increases in increments of 5 GB. |
| | | | 11<br>polardb.x8.4xlarge.2 | 32 cores, 256 GB | 12,000 | 240,000 | |
| | | | 11<br>polardb.x8.12xlarge.2 | 64 cores, 512 GB | 12,000 | 240,000 | 3,000 GB to 6,000 GB<br>The storage capacity increases in increments of 5 GB. |
| | | Dedicated host instance family | 11<br>polardb.x8.12xlarge.2 | 88 cores, 710 GB | 12,000 | 450,000 | 3,000 GB to 6,000 GB |
| | | | 11.2<br>polardb.x4.medium.2 | 2 cores, 8 GB | 400 | 10,000 | 250 GB |
| | | | 11.2<br>polardb.x8.medium.2 | 2 cores, 16 GB | 2,500 | 15,000 | |

| RDS edition | Engine version | Instance family | Instance type | CPU and memory capacity | Maximum number of connections | Maximum IOPS | Storage capacity |
|---|---|---|---|---|---|---|---|
| | 11.2 | Dedicated instance family | 11.2 polardb.x4.large.2 | 4 cores, 16 GB | 2,500 | 20,000 | 250 GB to 500 GB<br>The storage capacity increases in increments of 5 GB. |
| | | | 11.2 polardb.x8.large.2 | 4 cores, 32 GB | 5,000 | 30,000 | |
| | | | 11.2 polardb.x4.xlarge.2 | 8 cores, 32 GB | 5,000 | 40,000 | 500 GB to 1,000 GB<br>The storage capacity increases in increments of 5 GB. |
| | | | 11.2 polardb.x8.xlarge.2 | 8 cores, 64 GB | 10,000 | 60,000 | |
| | | | 11.2 polardb.x4.2xlarge.2 | 16 cores, 64 GB | 10,000 | 80,000 | 1,000 GB to 2,000 GB<br>The storage capacity increases in increments of 5 GB. |
| | | | 11.2 polardb.x8.2xlarge.2 | 16 cores, 128 GB | 12,000 | 120,000 | |
| | | | 11.2 polardb.x4.4xlarge.2 | 32 cores, 128 GB | 12,000 | 160,000 | 2,000GB to 3,000 GB |
| | | | 11.2 polardb.x8.4xlarge.2 | 32 cores, 256 GB | 12,000 | 240,000 | |

| RDS edition | Engine version | Instance family | Instance type | CPU and memory capacity | Maximum number of connections | Maximum IOPS | The storage capacity increases in increments of 5 GB. |
|---|---|---|---|---|---|---|---|
| | | | 11.2<br>polardb.x8.8xlarge.2 | 64 cores, 512 GB | 12,000 | 240,000 | 3,000 GB to 6,000 GB<br>The storage capacity increases in increments of 5 GB. |
| | | Dedicated host instance family | 11.2<br>polardb.x8.12xlarge.2 | 88 cores, 710 GB | 12,000 | 450,000 | |

# 12.KVStore for Redis

# 12.1. Product Introduction

## 12.1.1. 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 editions

| Edition | Overview |
| --- | --- |
| Community Edition instances | KVStore for Redis Community Edition is compatible with the data cache service of open source Redis engines. It supports master-replica instances, cluster instances, and read/write splitting instances. |
| Performance-enhanced instances of KVStore for Redis Enhanced Edition | KVStore for Redis Enhanced Edition 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. |

## 12.1.2. Enhanced Edition and supported commands

### 12.1.2.1. 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 |
| --- | --- |
| 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.<br>• Responds much faster when processing a large number of queries per second (QPS) compared with Redis-native databases.<br>• Maintains stable performance in high-concurrency scenarios and eliminates connection issues that are caused by traffic spikes during peak hours.<br>• 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.<br>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.<br>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.<br>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. |

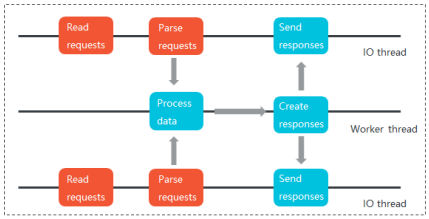
## Performance comparison

- 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<br>  | 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<br>  | To increase performance, each performance-enhanced instance of ApsaraDB for Redis runs multiple threads to process the tasks in these steps in parallel.<br>• I/O threads are used to read requests, send responses, and parse commands.<br>• Worker threads are used to process commands and timer events.<br>• Auxiliary threads are used to monitor the statuses of nodes and heartbeats.<br>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.<br>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.<br>**Note**<br>• The running speeds of threads are accelerated for common data structures, such as string, list, set, hash, zset, hyperloglog, geo, and extension structures.<br>• 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%.<br>• 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.

# 12.1.2.2. 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.<br>**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.<br>**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)
```

# 12.1.2.3. 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:

- 
- 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. |
| EXSETVER | 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. |

| Command | Syntax | Overview |
| --- | --- | --- |
| 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. |

| Parameter/option | Description |
| --- | --- |
| VER | The version number of the specified key.<br>○ If the specified key exists, the version number that is specified by this parameter is compared with the current version number.<br>■ If the version numbers match, the specified value is written to the key and the version number is increased by 1.<br>■ If this parameter does not match the current version number, an error message is returned.<br>○ 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. |

| Parameter/option | Description |
| --- | --- |
| VER | The version number of the specified key.<br>○ If the specified key exists, the version number that is specified by this parameter is compared with the current version number.<br>■ If the version numbers match, the value of the TairString is increased by num and the version number is increased by 1.<br>■ If this parameter does not match the current version number, an error message is returned.<br>○ 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. |
| 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.<br>○ If the specified key exists, the version number that is specified by this parameter is compared with the current version number.<br>■ If the version numbers match, the value of the TairString is increased by num and the version number is increased by 1.<br>■ If this parameter does not match the current version number, an error message is returned.<br>○ 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. |

| Parameter/option | Description |
| --- | --- |
| 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)
```

# 12.1.2.4. 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. |
| EXHMSET | EXHMSET <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. |
| 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. |

| Command | Syntax | Description |
| --- | --- | --- |
| 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. |
| 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.<br>**Note** To add a field that does not expire, you can run this command without the need to specify an expiration time. |

| Command | Syntax | Description |
| --- | --- | --- |
| 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.<br>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. |
| EXHGETWITHVER | EXHGETWITHVER <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. |
| EXHMGET | EXHMGET <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. |
| EXHMGETWITHVER | EXHMGETWITHVER <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. |

| Command | Syntax | Description |
| --- | --- | --- |
| 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 *noexp* 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. |
| EXHGETALL | EXHGETALL <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. |

| Command | Syntax | Description |
| --- | --- | --- |
| 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 immediately expires. If this option is not specified, the field does not expire. |
| EXAT | The absolute expiration time of the specified field. Unit: seconds. A value of 0 specifies that the field immediately expires. If this option is not specified, the field does not expire. |

| Parameter/option | Description |
| --- | --- |
| PX | The relative expiration time of the specified field. Unit: milliseconds. A value of 0 specifies that the field immediately expires. If this option is not specified, the field does not expire. |
| PXAT | The absolute expiration time of the specified field. Unit: milliseconds. A value of 0 specifies that the field immediately expires. If this option is not specified, 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.<br>○ If the specified field exists, the version number specified by this parameter is matched with the current version number:<br>■ If the version numbers match, the system continues running the command and increases the version number by 1.<br>■ If the version numbers do not match, an error message is returned.<br>○ 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 *EX, EXAT, PX, or PXAT* parameter, you can set the *NOACTIVE* 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.

## EXHMSET

- Syntax

  EXHMSET <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.<br>○ If the specified field exists, the version number specified by this parameter is matched with the current version number:<br>■ If the version numbers match, the system continues running the command and increases the version number by 1.<br>■ If the version numbers do not match, an error message is returned.<br>○ 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 *EX, EXAT, PX, or PXAT* parameter, you can set the *NOACTIVE* 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.<br>○ If the specified field exists, the version number specified by this parameter is matched with the current version number:<br>■ If the version numbers match, the system continues running the command and increases the version number by 1.<br>■ If the version numbers do not match, an error message is returned.<br>○ 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 *EX, EXAT, PX, or PXAT* parameter, you can set the *NOACTIVE* 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.<br>○ If the specified field exists, the version number specified by this parameter is matched with the current version number:<br>■ If the version numbers match, the system continues running the command and increases the version number by 1.<br>■ If the version numbers do not match, an error message is returned.<br>○ 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 *EX, EXAT, PX, or PXAT* parameter, you can set the *NOACTIVE* 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.<br>○ If the specified field exists, the version number specified by this parameter is matched with the current version number:<br>■ If the version numbers match, the system continues running the command and increases the version number by 1.<br>■ If the version numbers do not match, an error message is returned.<br>○ 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 *EX, EXAT, PX, or PXAT* parameter, you can set the *NOACTIVE* 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
  - -1: the specified key does not exist.
  - -2: the specified field does not exist.
  - The version number of the specified field is returned if the operation is successful.
  - 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
  - 0: the specified TairHash or field does not exist.
  - 1: the version number is specified.
  - 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 immediately expires. If this option is not specified, the field does not expire. |
| EXAT | The absolute expiration time of the specified field. Unit: seconds. A value of 0 specifies that the field immediately expires. If this option is not specified, the field does not expire. |
| PX | The relative expiration time of the specified field. Unit: milliseconds. A value of 0 specifies that the field immediately expires. If this option is not specified, the field does not expire. |
| PXAT | The absolute expiration time of the specified field. Unit: milliseconds. A value of 0 specifies that the field immediately expires. If this option is not specified, the field does not expire. |
| VER | The version number of the specified field.<br>○ If the specified field exists, the version number specified by this parameter is matched with the current version number:<br>■ If the version numbers match, the TairHash is increased by num and the version number is increased by 1.<br>■ If the version numbers do not match, an error message is returned.<br>○ If the value of the *VER* 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. |

| Parameter/option | Description |
| --- | --- |
| 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 *EX, EXAT, PX, or PXAT* parameter, you can set the *NOACTIVE* 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 immediately expires. If this option is not specified, 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 immediately expires. If this option is not specified, the field does not expire. |
| PX | The relative expiration time of the specified field. Unit: milliseconds. A value of 0 specifies that the field immediately expires. If this option is not specified, the field does not expire. |
| PXAT | The absolute expiration time of the specified field. Unit: milliseconds. A value of 0 specifies that the field immediately expires. If this option is not specified, the field does not expire. |
| VER | The version number of the specified field.<br>◦ If the specified field exists, the version number specified by this parameter is matched with the current version number:<br>▪ If the version numbers match, the TairHash is increased by num and the version number is increased by 1.<br>▪ If the version numbers do not match, an error message is returned.<br>◦ If the value of the *VER* 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 *EX, EXAT, PX, or PXAT* parameter, you can set the *NOACTIVE* 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 *noexp* parameter. When you set the *noexp* parameter,<br>○ the response time of the **EXHLEN** command is based on the size of the Tairhash, because the system scans all TairHashes.<br>○ 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:<br>>: specifies that the scan starts from the first field with the value of the key greater than the subkey.<br>>=: specifies that the scan starts from the first field with the value of the key greater than or equal to the subkey.<br><: specifies that the scan starts from the first field with the value of the key smaller than the subkey.<br><=: specifies that the scan starts from the first field with the value of the key smaller than or equal to the subkey.<br>==: specifies that the scan starts from the first field with the value of the key equal to the subkey.<br>^: specifies that the scan starts from the first field.<br>$: 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.

# 12.1.2.5. 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

- 
- 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 | `GIS.ADD <area> <polygonName> <polygonWkt> [<polygonName> <polygonWkt> ...]` | Adds one or more specified polygons to a specified area. The polygons are described in well-known text (WKT).<br>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 | `GIS.GET <area> <polygonName>` | Retrieves the WKT information about a specified polygon in an area. |
| GIS.GETALL | `GIS.GETALL <area> [WITHOUTWKT]` | Queries all polygons in a specified area. The names and WKT information of the polygons are returned. |
| GIS.DEL | `GIS.DEL <area> <polygonName>` | Deletes a specified polygon in an area. |

| Statement | Syntax | Description |
| --- | --- | --- |
| DEL | `DEL <key> [key ...]` | Deletes one or more TairGIS data structures. This is a Redis-native command. |
| GIS.CONTAINS | `GIS.CONTAINS <area> <polygonWkt> [WITHOUTWKT]` | Checks whether a polygon in a specified area consists of a specified point, line, or plane. |
| GIS.INTERSECTS | `GIS.INTERSECTS <area> <polygonWkt>` | Queries the intersection relationship between a polygon in a specified area and a specified point, line, or plane. |
| GIS.SEARCH | `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]` | Retrieves points within a sphere whose radius, latitude, and longitude are specified. |
| GIS.WITHIN | `GIS.WITHIN <area> <polygonWkt> [WITHOUTWKT]` | 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:<br>• POINT: the WKT information that describes a point, such as `'POINT (30 11)'`.<br>• LINESTRING: the WKT information that describes a line, such as `'LINESTRING (30 10, 40 40)'`.<br>• POLYGON: the WKT information that describes a polygon, such as `'POLYGON ((31 20, 29 20, 29 21, 31 31))'`.<br>Note MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, GEOMETRY, and COLLECTION are not supported. |
| WITHOUTWKT | Specifies whether to return the WKT information of polygons. If you run the GIS.GETALL, GIS.CONTAINS, GIS.SEARCH, or GIS.WITHIN command and specify the WITHOUTWKT 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 *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.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)"
```

# 12.1.2.6. 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. |
| 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.<br>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. |

| Parameter/option | Description |
| --- | --- |
| 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. |
| capacity | The initial capacity of the TairBloom filter. This parameter specifies the maximum number of items that can be added to the TairBloom filter.<br>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. |

| 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 filter exists, you do not need to specify this parameter.<br>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. |
| 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 |

## 12.1.2.7. 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 JavaScript Object Notation (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 and simplifies the retrieval of child elements.
- Supports conversion from the JSON format to the Extensible Markup Language (XML) or YAML Ain't Markup Language (YAML) format.

### Prerequisites

The commands described in this topic take effect only if the following conditions are met:

- A **performance-enhanced** instance of ApsaraDB for Redis Enhanced Edition is used.
- The TairDoc to be managed is stored on the performance-enhanced instance.

# Commands

## TairDoc commands

| Command | 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 a TairDoc path of a specified key. |
| JSON.DEL | JSON.DEL <key> [path] | Deletes JSON data from a TairDoc path of a specified key. If the path is not specified, the key is deleted. This command does not take effect if the key or path does not exist. |
| JSON.TYPE | JSON.TYPE <key> [path] | Retrieves the type of JSON data from a 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 both the JSON data and increased value must be of the int or double type. |
| 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 where you want to manage JSON data.<br>○ If the specified key does not exist, the path must be the root directory.<br>○ If the specified key and path exist, the specified JSON value overwrites the current JSON value in the path. |
| 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
  - OK: the operation is successful.
  - null: The operation fails. This occurs when you specify the NX or XX parameter.
  - Otherwise, an exception is returned.

- Examples

```
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 a 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 where 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. |
| ARRNAME | The tag that specifies an array element in an XML document. |

- Returned values
  - The JSON data stored in the path is returned if the operation is successful.
  - Otherwise, an exception is returned.

- Examples

```
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 a TairDoc path of a specified key. If the path is not specified, the key is deleted. This command does not take effect if the key or path does not exist.

- Parameters and options

| Parameter/option | Description |
| --- | --- |
| key | The key of the TairDoc that you want to manage. |
| path | The TairDoc path where you want to manage JSON data. |

- Returned values
  - 1: the operation is successful.
  - 0: the operation fails.
  - Otherwise, an exception is returned.

- Examples

```
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 a 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 where you want to manage JSON data. |

- Returned values
  - The type of JSON data is returned if the operation is successful. The type includes boolean, null, number, string, array, object, raw, reference, and const.
  - null: the specified key or path does not exist.
  - Otherwise, an exception is returned.
- Examples

```
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 both of the type of int or double.
- Parameters and options

| Parameter/option | Description |
| --- | --- |
| key | The key of the TairDoc that you want to manage. |
| path | The TairDoc path where you want to manage JSON data. |
| value | The increment to be added to the JSON data in the specified path. |

- Returned values
  - The increased value in the specified path is returned if the operation is successful.

- Otherwise, an exception is returned.

- Examples

```
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 where you want to manage JSON data. |
| json-string | The string to be appended to the specified path. |

- Returned values
  - The length of the increased value in the path is returned if the operation is successful.
  - -1: the specified key does not exist.
  - Otherwise, an exception is returned.

- Examples

```
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 where you want to manage JSON data. |

- Returned values
  - The length of the value in the path is returned if the operation is successful.
  - -1: the specified key does not exist.
  - Otherwise, an exception is returned.
- Examples

```
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 where you want to manage JSON data. |
| json | The JSON value to be inserted to a specified array. |

- Returned values
  - The number of elements in the array is returned if the operation is successful. The added elements are included.
  - -1: the specified key does not exist.
  - Otherwise, an exception is returned.

- Examples

```
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×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 where you want to manage JSON data. |
| index | The index of the array, which specifies the value to be removed. If you do not specify this parameter, the last value in the array is removed. A negative value specifies reverse numbering from the end of the array. |

- Returned values
  - The removed element is returned if the operation is successful.
  - An error message is returned if the array is empty: 'ERR array index outflow'.
  - Otherwise, an exception is returned.
- Examples

```
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×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 where you want to manage JSON data. |
| index | The index of the array, which specifies the value to be removed. If you do not specify 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 to a specified array. |

- Returned values
  - The number of elements in the array is returned if the operation is successful. The added elements are included.
  - An error message is returned if the array is empty: 'ERR array index outflow'.
  - Otherwise, an exception is returned.
- Examples

```
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 where you want to manage JSON data. |

- Returned values
  - The length of the queried array is returned if the operation is successful.
  - -1: the specified key does not exist.
  - Otherwise, an exception is returned.
- Examples

```
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 where 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
  - The length of the trimmed array is returned if the operation is successful.
  - -1: the specified key does not exist.

- Otherwise, an exception is returned.

- Examples

```
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 |
| --- | --- |
| 127.0.0.1:6379> JSON.SET doc . '{"foo": "bar", "baz" : [1,2,3]}'<br>OK<br>127.0.0.1:6379> JSON.GET doc .foo<br>"bar"<br>127.0.0.1:6379> JSON.GET doc .baz[0]<br>1 | 127.0.0.1:6379> JSON.SET doc "" '{"foo": "bar", "baz" : [1,2,3]}'<br>OK<br>127.0.0.1:6379> JSON.GET doc /foo<br>"bar"<br>127.0.0.1:6379> JSON.GET doc /baz/0<br>1 |

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 |
| Multiple elements in a path | .a['b.c'] | /a/b.c |

# 12.1.3. 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.

## 12.1.4. System architecture and components

This topic describes the system architecture and components of KVStore for Redis.

Architecture

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.

# 12.1.5. Features

KVStore for Redis supports multiple architectures, persistent data storage, high availability, auto scaling, and intelligent operations and maintenance (O&M).

## Flexible architectures

| Architecture | Description | Scenario |
| --- | --- | --- |
| Standard instance | The master node serves your workloads and the replica node stays in hot standby mode to ensure high availability. If the master node fails, the system switches the workloads to the replica node. This mechanism guarantees the high availability for your workloads. | • Support for more native Redis features.<br>• Persistent storage in KVStore for Redis instances.<br>• Stable query rate on a single node of VKStore for Redis<br>• Use of simple Redis commands, when only a few sorting and computing commands are required. |
| Cluster instance | • A cluster instance contains proxy servers, data shards, and config servers. You can scale out a cluster instance by increasing the number of data shards.<br>• A cluster master-replica instance contains multiple data shards. Each data shard works in a high-availability (HA) architecture in which a master node and a replica node are deployed on different hosts. If the master node is faulty, the cluster master-replica instance fails over to the replica node to ensure high service availability. | • Large data volume.<br>• High queries per second (QPS).<br>• High-throughput and high-performance applications. |

| Architecture | Description | Scenario |
| --- | --- | --- |
| Read/write splitting instance | • A read/write splitting instance contains proxy servers, master and replica nodes, and read replicas.<br>• Read replicas support chained replication. This allows you to scale out read replicas to increase the read capacity. | • High QPS scenarios, such as data hotspots.<br>• Support for more native Redis features. You can use read/write splitting instances to overcome the limits of cluster instances.<br>**Note** Data synchronization to read replicas has a latency. Therefore, in scenarios that require high data consistency, we recommend that you use cluster instances instead of read/write splitting instances. |

## Data security

| Item | Description |
| --- | --- |
| Data backup and restoration | • Data backup: KVStore for Redis uses Redis database backup (RDB) snapshots to persist data. One backup is automatically created a day from the replica node based on the default backup policy. You can modify the automatic backup policy or manually create a temporary backup.<br>• Data restoration: You can restore data from a specified backup set to the current instance or a new instance. When you restore data to a new instance, the data in the new instance is the same as that in the backup set. This can be used in scenarios such as data restoration, rapid business deployment, or data verification.<br>• Backup file downloading: Backup files of KVStore for Redis are retained for seven days. If you want to retain backup files for more than seven days due to regulatory or security requirements, you can download the backup files to your on-premises machine. |
| Multi-layer network security protection | • KVStore for Redis supports VPCs. VPCs are logically isolated from each other at Layer 2 to provide higher security and higher performance.<br>• Anti-DDoS monitors and protects against Distributed-Denial-of-Service (DDoS) attacks.<br>• You can configure more than 1,000 IP address whitelists to block malicious IP addresses. You must complete whitelist settings before you connect to the instance.<br>• KVStore for Redis provides password authentication to ensure secure and reliable access. For an instance, you can create up to 20 accounts. You can grant appropriate permissions to accounts based on business requirements. This allows you to flexibly manage instances and prevent accidental operations.<br>• SSL encryption is supported. You can install an SSL certificate to your instance after SSL is enabled. This helps you encrypt network connections at the transport layer, improving the security of communication data and ensuring data integrity. |

| Item | Description |
| --- | --- |
| In-depth kernel optimization | Alibaba Cloud performed an in-depth engine optimization on the Redis source code to prevent memory shortage, fix security vulnerabilities, and protect your business. |

## High availability

| Item | Description |
| --- | --- |
| Master-replica architecture | Shards use the master-replica architecture. The master and replica nodes implement real-time data synchronization by using RDB and AOF. The master node serves your workloads and the replica node stays in hot standby mode to ensure high availability. If the master node fails, the system switches the workloads to the replica node. This mechanism guarantees the high availability for your workloads. |
| Redundancy and automatic detection | • A redundancy design is adopted for each system component to eliminate the risk of single points of failure.<br>• The system automatically detects hardware failures. In the case of failures, the system performs a failover and restores services within seconds. |

## Scalability

| Item | Description |
| --- | --- |
| Capacity scaling | If the performance of the instance becomes insufficient or excessive after a KVStore for Redis instance is created, you can change the architecture or memory specifications of the instance. This can meet the performance and capacity requirements in different scenarios. |
| Performance scaling | |

## Intelligent O&M

| Item | Description |
| --- | --- |
| Performance monitoring | KVStore for Redis provides abundant performance monitoring metrics such as CPU utilization and connections. You can query the monitoring during a specified period of time in the past month. This helps you check the health status and trace problems of KVStore for Redis. |
| Visualized management | The web-based visual management console provides rich management features such as data backup and parameter settings. You can manage instances in a convenient and visualized manner. |
| Database kernel version management | KVStore Redis continuously optimizes the kernel and fixes security vulnerabilities to improve service stability. You can upgrade the minor version and kernel version in the console. |

# 12.1.6. Scenarios

## Gaming 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, KVStore for Redis 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.

# 12.1.7. 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 you set the number of elements in a set to a value 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 you set the number of elements in a sorted set to a value 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 you set the number of elements in a hash table to a value less than 8,192. The value length is 1 MB or less. |
| Number of databases | A single instance supports a maximum of 256 databases. We recommend that you use multiple instances to support more databases. |
| Policy to delete expired data | • Two expiration policies are supported, which are active expiration and passive expiration. In active expiration, the system periodically detects and deletes expired keys in the background. This policy does not ensure timeliness.<br>• In passive expiration, the system detects and deletes expired keys when you access these keys.<br>**Note** In versions earlier than Redis 4.0, network jitter may occur due to high resource consumption caused by the deletion of large keys. |
| Mechanism to recycle idle connections | KVStore for Redis does not automatically recycle idle connections to KVStore for Redis. You can manage the connections. |
| Policy for data persistence | KVStore for Redis uses the AOF_FSYNC_EVERYSEC method and runs the fysnc command at an interval of 1 second. |

# 12.1.8. 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. |

# 12.1.9. Instance types

KVStore for Redis provides instances of the Community Edition and performance-enhanced instances of the Enhanced Edition (Tair). Performance-enhanced instances of the Enhanced Edition (Tair) support the standard, cluster, and read/write splitting architectures. Instances of the Community Edition support the standard and cluster architectures. This topic describes the specifications of different instance types, such as memory capacity, maximum number of connections, and maximum bandwidth value.

## Bandwidth description

- 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 cluster instance is 2,048 MB/s. After the upper limit is reached, the bandwidth cannot be increased even if the number of shards is increased.
- 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 tables is the internal bandwidth of the KVStore for Redis instance. The Internet bandwidth is determined by the internal bandwidth and is limited by the bandwidth of the connection between the KVStore for Redis instance and the client. We recommend that you

connect to the instance over an internal network to maximize performance.

## Performance-enhanced instances of the KVStore for Redis Enhanced Edition (Tair)

### Standard master-replica instances

| Specification | InstanceClass value (used in API operations) | Number of I/O threads | Number of connections per second | Maximum number of connections | Total maximum bandwidth value (MB/s) |
|---|---|---|---|---|---|
| 1 GB master-replica performance-enhanced instance | redis.amber.master.small.multithread | 6 | 10,000 | 30,000 | 96 |
| 2 GB master-replica performance-enhanced instance | redis.amber.master.mid.multithread | 6 | 10,000 | 30,000 | 96 |
| 4 GB master-replica performance-enhanced instance | redis.amber.master.stand.multithread | 6 | 10,000 | 30,000 | 96 |
| 8 GB master-replica performance-enhanced instance | redis.amber.master.large.multithread | 6 | 10,000 | 30,000 | 96 |
| 16 GB master-replica performance-enhanced instance | redis.amber.master.2xlarge.multithread | 6 | 10,000 | 30,000 | 96 |
| 32 GB master-replica performance-enhanced instance | redis.amber.master.4xlarge.multithread | 6 | 10,000 | 30,000 | 96 |
| 64 GB master-replica performance-enhanced instance | redis.amber.master.8xlarge.multithread | 6 | 10,000 | 30,000 | 96 |

### Cluster instances

| Specification | InstanceClass value (used in API operations) | Number of I/O threads | Number of shards | Number of proxy nodes | Number of connections per second | Maximum number of 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 | 6 | 50,000 | 50,000 | 96 | 192 |
| 4 GB cluster performance-enhanced instance (2 shards) | redis.amber.logic.sharding.2g.2db.0rodb.6proxy.multithread | 6 | 2 | 6 | 50,000 | 50,000 | 96 | 192 |
| 8 GB cluster performance-enhanced instance (4 shards) | redis.amber.logic.sharding.2g.4db.0rodb.12proxy.multithread | 6 | 4 | 12 | 50,000 | 120,000 | 96 | 384 |
| 16 GB cluster performance-enhanced instance (8 shards) | redis.amber.logic.sharding.2g.8db.0rodb.24proxy.multithread | 6 | 8 | 24 | 50,000 | 240,000 | 96 | 768 |
| 32 GB cluster performance-enhanced instance (8 shards) | redis.amber.logic.sharding.4g.8db.0rodb.24proxy.multithread | 6 | 8 | 24 | 50,000 | 240,000 | 96 | 768 |
| 64 GB cluster performance-enhanced instance (8 shards) | redis.amber.logic.sharding.8g.8db.0rodb.24proxy.multithread | 6 | 8 | 24 | 50,000 | 240,000 | 96 | 768 |
| 32 GB cluster performance-enhanced instance (16 shards) | redis.amber.logic.sharding.2g.16db.0rodb.48proxy.multithread | 6 | 16 | 48 | 50,000 | 480,000 | 96 | 1,536 |
| 64 GB cluster performance-enhanced (16 shards) | redis.amber.logic.sharding.4g.16db.0rodb.48proxy.multithread | 6 | 16 | 48 | 50,000 | 480,000 | 96 | 1,536 |

| Specification | InstanceClass value (used in API operations) | Number of I/O threads | Number of shards | Number of proxy nodes | Number of connections per second | Maximum number of 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 | 48 | 50,000 | 480,000 | 96 | 1,536 |
| 256 GB cluster performance-enhanced instance (16 shards) | redis.amber.logic.sharding.16g.16db.0rodb.48proxy.multithread | 6 | 16 | 48 | 50,000 | 480,000 | 96 | 1,536 |
| 512 GB cluster performance-enhanced instance (32 shards) | redis.amber.logic.sharding.16g.32db.0rodb.96proxy.multithread | 6 | 32 | 96 | 50,000 | 500,000 | 96 | 2,048 |
| 1,024 GB cluster performance-enhanced instance (64 shards) | redis.amber.logic.sharding.16g.64db.0rodb.192proxy.multithread | 6 | 64 | 192 | 50,000 | 500,000 | 96 | 2,048 |
| 2,048 GB cluster performance-enhanced instance (128 shards) | redis.amber.logic.sharding.16g.128db.0rodb.384proxy.multithread | 6 | 128 | 384 | 50,000 | 500,000 | 96 | 2,048 |
| 4,096 GB cluster performance-enhanced instance (256 shards) | redis.amber.logic.sharding.16g.256db.0rodb.768proxy.multithread | 6 | 256 | 768 | 50,000 | 500,000 | 96 | 2,048 |
| 8,192 GB cluster performance-enhanced instance (256 shards) | redis.amber.logic.sharding.32g.256db.0rodb.768proxy.multithread | 6 | 256 | 768 | 50,000 | 500,000 | 96 | 2,048 |

## Community Edition instances

### Standard master-replica instances

| Specification | InstanceClass value (used in API operations) | Maximum number of connections | Total maximum bandwidth value (MB/s) | Processing capability |
|---|---|---|---|---|
| 1GB master-replica instance | redis.master.small.default | 10,000 | 10 | 1 core |
| 2 GB master-replica instance | redis.master.mid.default | 10,000 | 16 | 1 core |
| 4 GB master-replica instance | redis.master.stand.default | 10,000 | 24 | 1 core |
| 8 GB master-replica instance | redis.master.large.default | 10,000 | 24 | 1 core |
| 16 GB master-replica instance | redis.master.2xlarge.default | 10,000 | 32 | 1 core |
| 32 GB master-replica instance | redis.master.4xlarge.default | 10,000 | 32 | 1 core |
| 64 GB master-replica instance | redis.master.8xlarge.default | 10,000 | 48 | 1 core |

### Cluster instances

| Specification | InstanceClass value (used in API operations) | Number of shards | Maximum number of connections | Total maximum bandwidth value (MB/s) | Processing capability |
|---|---|---|---|---|---|
| 16 GB cluster instance | redis.logic.sharding.2g.8db.0rodb.8proxy.default | 8 | 80,000 | 768 | 8 cores |
| 32 GB cluster instance | redis.logic.sharding.4g.8db.0rodb.8proxy.default | 8 | 80,000 | 768 | 8 cores |
| 64 GB cluster instance | redis.logic.sharding.8g.8db.0rodb.8proxy.default | 8 | 80,000 | 768 | 8 cores |
| 128 GB cluster instance | redis.logic.sharding.8g.16db.0rodb.16proxy.default | 16 | 160,000 | 1,536 | 16 cores |
| 256 GB cluster instance | redis.logic.sharding.16g.16db.0rodb.16proxy.default | 16 | 160,000 | 1,536 | 16 cores |
| 512 GB cluster instance | redis.logic.sharding.16g.32db.0rodb.32proxy.default | 32 | 320,000 | 2,048 | 32 cores |
| 1 TB cluster instance | redis.logic.sharding.16g.64db.0rodb.64proxy.default | 64 | 500,000 | 2,048 | 64 cores |

| Specification | InstanceClass value (used in API operations) | Number of shards | Maximum number of connections | Total maximum bandwidth value (MB/s) | Processing capability |
|---|---|---|---|---|---|
| 2 TB cluster instance | redis.logic.sharding.16g.128db.0rodb.128proxy.default | 128 | 500,000 | 2,048 | 128 cores |
| 4 TB cluster instance | redis.logic.sharding.16g.256db.0rodb.256proxy.default | 256 | 500,000 | 2,048 | 256 cores |

### Read/write splitting instances

| Specification | InstanceClass value (used in API operations) | Number of read replicas | Number of connections per second | Maximum number of concurrent connections | Total maximum bandwidth value (MB/s) | QPS |
|---|---|---|---|---|---|---|
| 1 GB read/write splitting instance (1 shard, 1 read replica) | redis.logic.splitrw.small.1db.1rodb.4proxy.default | 1 | 20,000 | 20,000 | 96 | 200,000 |
| 2 GB read/write splitting instance (1 shard, 1 read replica) | redis.logic.splitrw.mid.1db.1rodb.4proxy.default | 1 | 20,000 | 20,000 | 192 | 200,000 |
| 4 GB read/write splitting instance (1 shard, 1 read replica) | redis.logic.splitrw.stand.1db.1rodb.4proxy.default | 1 | 20,000 | 20,000 | 192 | 200,000 |
| 8 GB read/write splitting instance (1 shard, 1 read replica) | redis.logic.splitrw.large.1db.1rodb.4proxy.default | 1 | 20,000 | 20,000 | 192 | 200,000 |
| 16 GB read/write splitting instance (1 shard, 1 read replica) | redis.logic.splitrw.2xlarge.1db.1rodb.4proxy.default | 1 | 20,000 | 20,000 | 192 | 200,000 |
| 32 GB read/write splitting instance (1 shard, 1 read replica) | redis.logic.splitrw.4xlarge.1db.1rodb.4proxy.default | 1 | 20,000 | 20,000 | 192 | 200,000 |
| 64 GB read/write splitting instance (1 shard, 1 read replica) | redis.logic.splitrw.8xlarge.1db.1rodb.4proxy.default | 1 | 20,000 | 20,000 | 192 | 200,000 |

| Specification | InstanceClass value (used in API operations) | Number of read replicas | Number of connections per second | Maximum number of concurrent connections | Total maximum bandwidth value (MB/s) | QPS |
|---|---|---|---|---|---|---|
| 1 GB read/write splitting instance (1 shard, 3 read replicas) | redis.logic.splitrw.small.1db.3rodb.4proxy.default | 3 | 40,000 | 40,000 | 192 | 400,000 |
| 2 GB read/write splitting instance (1 shard, 3 read replicas) | redis.logic.splitrw.mid.1db.3rodb.4proxy.default | 3 | 40,000 | 40,000 | 384 | 400,000 |
| 4 GB read/write splitting instance (1 shard, 3 read replicas) | redis.logic.splitrw.stand.1db.3rodb.4proxy.default | 3 | 40,000 | 40,000 | 384 | 400,000 |
| 8 GB read/write splitting instance (1 shard, 3 read replicas) | redis.logic.splitrw.large.1db.3rodb.4proxy.default | 3 | 40,000 | 40,000 | 384 | 400,000 |
| 16 GB read/write splitting instance (1 shard, 3 read replicas) | redis.logic.splitrw.2xlarge.1db.3rodb.4proxy.default | 3 | 40,000 | 40,000 | 384 | 400,000 |
| 32 GB read/write splitting instance (1 shard, 3 read replicas) | redis.logic.splitrw.4xlarge.1db.3rodb.4proxy.default | 3 | 40,000 | 40,000 | 384 | 400,000 |
| 64 GB read/write splitting instance (1 shard, 3 read replicas) | redis.logic.splitrw.8xlarge.1db.3rodb.4proxy.default | 3 | 40,000 | 40,000 | 384 | 400,000 |

# 13.ApsaraDB for MongoDB

# 13.1. Product Introduction

## 13.1.1. ApsaraDB for MongoDB

ApsaraDB for MongoDB is a high-performance document database service that is fully compatible with the MongoDB protocol. Based on the Apsara distributed system and a highly reliable storage engine, ApsaraDB for MongoDB provides features such as high availability, elastic scaling, read/write splitting, high data security, backup and restoration, intelligent O&M, and online database management.

## 13.1.1.1. What is ApsaraDB for MongoDB?

ApsaraDB for MongoDB is a MongoDB-compatible document-oriented database service that is developed based on the Apsara system and a high-reliability storage engine. ApsaraDB for MongoDB uses a multi-node architecture to ensure high availability and supports elastic scaling, disaster recovery, backup and restoration, and performance optimization.

### Data structure

MongoDB is a document-oriented NoSQL database. MongoDB stores data in JSON-like documents that consist of field-value pairs. Example:

```
{
   name:"John",
   sex:"male",
   age:30
}
```

### Storage structure

The storage structure of MongoDB is different from that of conventional relational databases. Data in MongoDB is organized at the following levels:

- Document

  Documents are the basic unit of data in MongoDB. A document consists of BSON key-value pairs and is equivalent to a row in a relational database.

- Collection

  A collection can contain multiple documents. Collections are equivalent to tables in a relational database.

- Database

  A database can contain multiple collections. You can create multiple databases in MongoDB. Databases are equivalent to relational databases.

### Instance architectures

ApsaraDB for MongoDB provides the following instance architectures:

- Replica set instances

  An ApsaraDB for MongoDB replica set instance consists of a primary node, one or more secondary nodes, and a hidden node.

  - Primary node: processes all read and write operations.
  - Secondary node: synchronizes data from the primary node. If the primary node fails, a secondary node becomes the new primary node to ensure high availability.
  - Hidden node: ensures high availability of the instance. If a secondary node fails, the hidden node becomes the secondary node.

    > **Note** The hidden node is used only to ensure high availability. It is invisible to users.

- Sharded cluster instances

  An ApsaraDB for MongoDB sharded cluster instance consists of mongos, shard, and Configserver nodes.

  - Mongos node: routes queries and write operations to the corresponding shard node.
  - Shard node: stores database data.
  - Configserver node: stores data of shard nodes.

## Deployment suggestions

When you deploy an ApsaraDB for MongoDB instance, you can consider the following aspects:

- Regions and zones

  You can select a region and zone based on your location, the availability of Alibaba Cloud services, your application availability requirements, and whether internal network communication is required.

  > **Note** A region is an Alibaba Cloud data center. A zone is a physical area within a region that has its own independent power supply and network.

  For example, if your application is deployed on an Elastic Compute Service (ECS) instance and requires an ApsaraDB for MongoDB instance to serve as its database, you must select the same region and zone as the ECS instance when you create your ApsaraDB for MongoDB instance.

  > **Note**
  >
  > - An ECS instance and an ApsaraDB for MongoDB instance within the same zone can be connected by using an internal network with minimal network latency.
  > - The region and zone determine the physical location of an ApsaraDB for MongoDB instance. You cannot change the region of an ApsaraDB for MongoDB instance after the instance is created.

- Network planning

A virtual private cloud (VPC) helps you build an isolated network environment on Apsara Stack. You can customize route tables, CIDR blocks, and gateways within a VPC. We recommend that you select VPC for improved security.

- Security policies

  ApsaraDB for MongoDB provides comprehensive security measures to improve data security. You can ensure database security by means of zone-disaster recovery, audit logs, network isolation, whitelists, password authentication, SSL encryption, and Transparent Data Encryption (TDE).

# 13.1.1.2. Benefits

ApsaraDB for MongoDB is a MongoDB-compatible document-oriented database service that is developed based on the Apsara system and a high-reliability storage engine. ApsaraDB for MongoDB ensures high availability, supports elastic scaling, and delivers security.

## High availability

- High-availability architecture, zone-disaster recovery, and automatic backup to ensure business availability
  - ApsaraDB for MongoDB provides the three-node replica set and sharded cluster architectures. Multiple data nodes are deployed on various physical servers. When a node in an instance fails, other nodes automatically synchronize data to ensure the high availability of the instance.
  - ApsaraDB for MongoDB allows you to create dual-zone instances. When a zone for a dual-zone instance becomes unavailable due to unexpected events, the data in the instance can be synchronized in the other zone to ensure the continued availability of the instance.
  - ApsaraDB for MongoDB provides the automatic backup feature. The system automatically backs up data and uploads the data to Object Storage Service (OSS) during the specified time period. This improves disaster recovery capabilities and reduces consumed disk capacity. Backup files can be used to restore instance data to their source instance and prevent irreversible effects on business data caused by accidental changes and other errors.
- Primary/secondary failover to ensure service availability

  ApsaraDB for MongoDB provides the primary/secondary failover feature. When a node of an instance fails, the system triggers a primary/secondary failover to ensure availability of the instance.

## Elastic scaling

- Flexible configuration changes to meet business requirements

  ApsaraDB for MongoDB allows you to change instance configurations. Multiple instance types are available for configuration change. You can change instance configurations as your business needs change.
- Multiple chip architectures for hybrid deployment

  ApsaraDB for MongoDB provides a variety of chip architectures such as x86 and ARM. You can select a chip architecture based on your needs for scenarios such as scale-out events, disaster recovery, and hybrid deployment.

## Security

- Pre-protection

  ApsaraDB for MongoDB provides the DDoS mitigation feature. ApsaraDB for MongoDB monitors inbound traffic in real time, filters source IP addresses to scrub large amounts of malicious traffic, and triggers blackhole filtering if traffic scrubbing becomes ineffective.

- In-event protection
  - Configuration of IP address whitelists for enhanced database access security

    IP addresses or CIDR blocks used to access databases can be added to a whitelist of the instance to ensure security and stability of databases.

  - SSL encryption for enhanced link security

    SSL encryption can encrypt network connections at the transport layer to improve data security and ensure data integrity. After SSL encryption is enabled, you can install SSL certificates issued by certification authorities (CAs) on your application to improve link security.

  - TDE for improved data security

    Transparent Data Encryption (TDE) is used to encrypt data before the data is written from data files to a disk and decrypt data before the data is read from a disk and written to the memory. TDE does not increase the sizes of data files. You can use TDE without the need to modify the configuration data of your application. You can enable TDE for an instance to encrypt instance data and improve data security.

- Post-auditing

  ApsaraDB for MongoDB automatically stores audit logs in Log Service and allows you to download these logs from Log Service. This facilitates the long-term storage and management of audit logs.

## Intelligent O&M

- Comprehensive monitoring to help O&M personnel understand the running status of instances

  ApsaraDB for MongoDB provides a variety of performance monitoring metrics such as CPU utilization, memory usage, and disk usage for you to view the running status of your instances in real time.

- Performance optimization to ensure stability, security, and efficiency of databases

  ApsaraDB for MongoDB allows you to view and customize instance performance trends and view performance, storage, and slow query logs of instances in real time. This helps you eliminate service failures caused by manual operations and ensure the stability, security, and efficiency of databases.

## Network isolation

ApsaraDB for MongoDB uses Virtual Private Cloud (VPC) to implement advanced network access control. VPC and IP address whitelists greatly improve the security of ApsaraDB for MongoDB instances.

A VPC can help you build an isolated network environment by using underlying network protocols. You can resolve resource conflicts by customizing route tables, IP addresses, and gateways in VPCs.

By default, ApsaraDB for MongoDB instances deployed in a VPC can be accessed only by the Elastic Compute Service (ECS) instances in the same VPC. If necessary, you can apply for a public IP address to allow access requests from the Internet (not recommended). Before you apply for a public IP address, you must configure a whitelist. For example, you can allow access requests from elastic IP addresses (EIPs) of ECS instances and the Internet egress of your data center.

### Online management of databases

Data Management (DMS) is an integrated and visualized database solution that offers data management, structure management, user authorization, security auditing, data trend analysis, data tracking, business intelligence (BI) charts, performance optimization, and server management. You can use DMS to log on to the ApsaraDB for MongoDB console and obtain a list of ApsaraDB for MongoDB instances for remote access and online management.

## 13.1.1.3. Scenarios

ApsaraDB for MongoDB is a MongoDB-compatible document-oriented database service that can store large amounts of data and provides features such as security auditing and backup and restoration. It is widely used in scenarios such as IoT and gaming.

### Read/write splitting

ApsaraDB for MongoDB uses a three-node replica set architecture to ensure high availability. Three data nodes are deployed on different physical servers and automatically synchronize data. The primary and secondary nodes are configured with different endpoints. MongoDB drivers allocate read/write requests to these nodes.

### Flexible business scenarios

ApsaraDB for MongoDB has no schema and is suitable for startups that do not want to go through the hassles of changing table schemas. You can store structured data that has a fixed schema in ApsaraDB RDS, data that has flexible schemas in ApsaraDB for MongoDB, and hot data in ApsaraDB for Redis or ApsaraDB for Memcache. This ensures that business data can be stored and retrieved in an efficient manner to reduce data storage costs.

### Mobile apps

ApsaraDB for MongoDB supports two-dimensional spatial indexes. Therefore, ApsaraDB for MongoDB can provide support for location-based apps. ApsaraDB for MongoDB uses a dynamic storage method, which is suitable for storing heterogeneous data from multiple systems and meets the needs of mobile apps.

### IoT scenarios

- ApsaraDB for MongoDB provides high performance and allows data to be asynchronously written.
- Three types of components are available in ApsaraDB for MongoDB sharded cluster instances: mongos, shard, and Configserver nodes. You can configure the number and specifications of

mongos and shard nodes when you create sharded cluster instances to provide different levels of performance. This makes ApsaraDB for MongoDB suitable for IoT scenarios that involve highly concurrent write operations.

- ApsaraDB for MongoDB provides the secondary index feature for dynamic queries. It can use the MapReduce aggregation framework of MongoDB to conduct multidimensional data analysis.

## Applications in various fields

- Gaming

  ApsaraDB for MongoDB can be used as a database service for game servers to store user information. The in-game equipment and credits of players are directly stored in the form of an embedded document to facilitate queries and updates.

- Logistics

  ApsaraDB for MongoDB can store order information. The order status is constantly updated during the shipping process and is stored in the form of an embedded array in ApsaraDB for MongoDB. You can read all the changes in an order in a straightforward manner by performing a single query.

- Social networking
  - ApsaraDB for MongoDB can store the information of users and their published WeChat moments. Geographical location indexes can be used to search nearby people and places.
  - Additionally, ApsaraDB for MongoDB is suitable for storing chat history because ApsaraDB for MongoDB provides rich query abilities and is fast in both writing and reading.

- Live video streaming

  ApsaraDB for MongoDB can store user information and gift information.

# 14.AnalyticDB for PostgreSQL

# 14.1. Product Introduction

## 14.1.1. 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.

### 14.1.1.1. Features

This topic describes the features of AnalyticDB for PostgreSQL.

#### Distributed architecture

AnalyticDB for PostgreSQL is built on a massively parallel processing (MPP) architecture. Data is evenly distributed across nodes by hash value or RANDOM function, and is analyzed and computed in parallel. As your data volume increases, you can add nodes to ensure that storage and computing capabilities can meet query response requirements.

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) capabilities for aggregate analysis. AnalyticDB for PostgreSQL supports B-tree indexes, bitmap 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, workloads are automatically switched to the secondary coordinator node.

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

## Data synchronization methods and tools

You can use Data Transmission Service (DTS) or DataWorks 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) tools such as Quick BI, DataV, Tableau, and FineReport, and ETL tools such as Informatica and Kettle.

## Data security

AnalyticDB for PostgreSQL supports the configuration of allowlists. You can add up to 1,000 IP addresses of servers to an allowlist 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 based on the IP address. 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 a variety of indexes, including B-tree indexes, bitmap 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 partitioned tables and a variety of partition-related operations such as ADD, DROP, RENAME, TRUNCATE, EXCHANGE, and SPLIT.
- Supports a variety of data types. The following table describes the supported data types.

| Parameter | Alias | Storage size | Range | Description |
| --- | --- | --- | --- | --- |
| bigint | int8 | 8 bytes | -9223372036854775808 to 9223372036854775807 | An integer within a large range. |

| Parameter | Alias | Storage size | Range | Description |
| --- | --- | --- | --- | --- |
| 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 to 294277 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<br>float | 8 bytes | Precise to 15 decimal digits | A variable precision. This data type is an inexact type. |
| 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. |

| Parameter | Alias | Storage size | Range | Description |
| --- | --- | --- | --- | --- |
| interval [ (p) ] | None | 12 bytes | -178000000 years to 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] to 24:00:00[.000000] | The time of a day without the time zone. |

| Parameter | Alias | Storage size | Range | Description |
| --- | --- | --- | --- | --- |
| time [ (p) ] with time zone | timetz | 12 bytes | 00:00:00+1359 to 24:00:00-1359 | The time of a day with the time zone. |
| timestamp [ (p) ] [ without time zone ] | None | 8 bytes | 4713 BC to 294277 AD | The date and time without the time zone. |
| timestamp [ (p) ] with time zone | timestamptz | 8 bytes | 4713 BC to 294277 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. |

# 14.1.1.2. Benefits

This topic describes the benefits of AnalyticDB for PostgreSQL.

- Real-time analysis

  Built on a massively parallel processing (MPP) architecture that supports horizontal scaling and can respond to queries on petabytes of data within seconds. Compared with traditional databases, AnalyticDB for PostgreSQL supports vector computing 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 atomicity, consistency, isolation, and durability (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 improve the security of your data.

- Ease of use

  Supports a variety of 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 extract, transform, and load (ETL) tools online.

- Ultra-high performance

Supports row store, column store, and a variety of 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 Object Storage Service (OSS).

- Flexible scalability

  Supports on-demand scale-out for more CPU, memory, and storage resources to improve online analytical processing (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 by using multiple instances. Tenant tasks are submitted to queues on different instances for execution. Tenant resources are isolated by isolating AnalyticDB for PostgreSQL instances.

- Permission management

  Allows you to manage tenants in a centralized manner and dynamically configure tenant resources. 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 server clusters and resource pools.

- Mixed deployment

  Supports deployment with mixed x86 and ARM server clusters.

## 14.1.1.3. Scenarios

This topic describes the use scenarios of AnalyticDB for PostgreSQL.

AnalyticDB for PostgreSQL is suitable for the following online analytical processing (OLAP) tasks:

- Extract, transform, and load (ETL) tasks 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 to explore, warehouse, and update data in real time:

- 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 indexes to obtain results within milliseconds. Supported indexes include B-tree indexes, bitmap indexes, and hash indexes.
- Supports distributed transactions, standard database isolation levels, and hybrid transaction/analytical processing (HTAP).

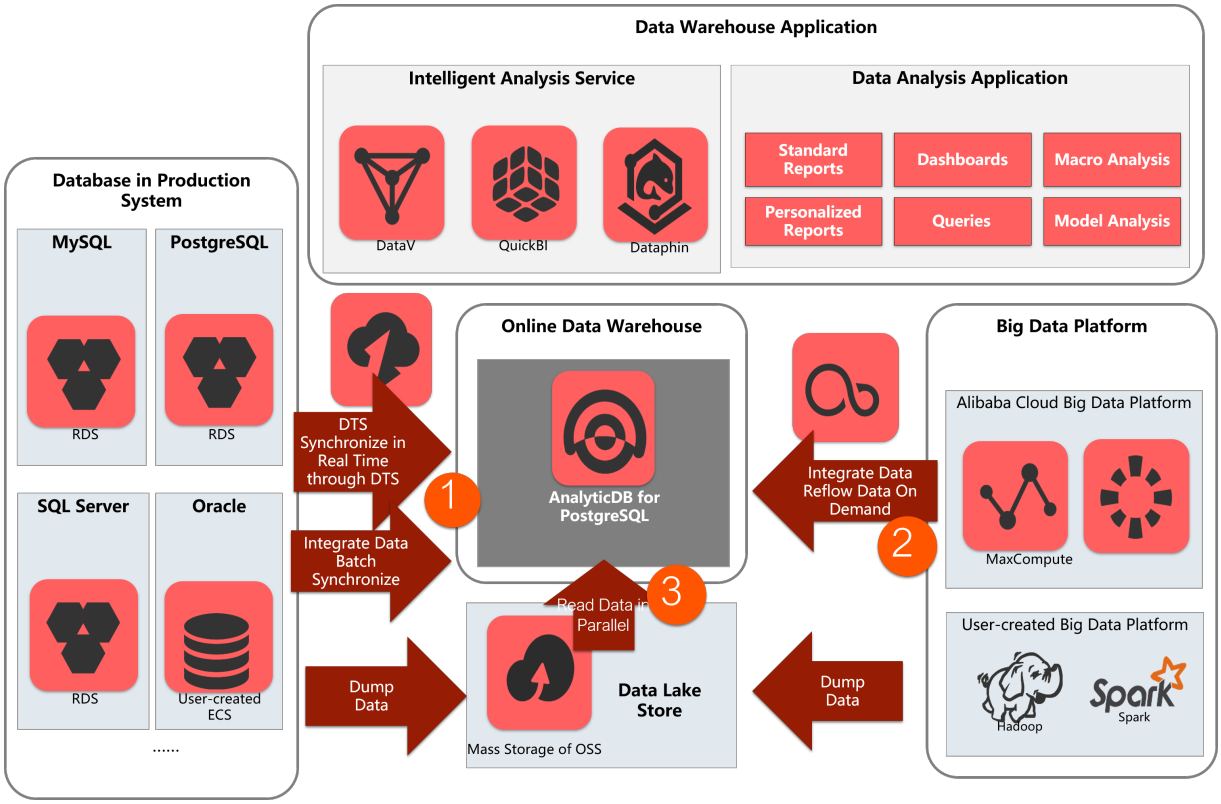
- Multi-model data analysis

  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.

## Use scenarios

The following figure shows the use scenarios of AnalyticDB for PostgreSQL.




- Enterprise data warehousing service

Data in your enterprise systems can be synchronized from your databases to AnalyticDB for PostgreSQL in real time by using Data Transmission Service (DTS) or in batches by using Data Integration. Supported databases include cloud databases such as ApsaraDB RDS for MySQL, ApsaraDB RDS for PostgreSQL, and PolarDB and traditional databases such as Oracle and SQL Server. 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 to implement 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.

# 15.Data Transmission Service (DTS)

# 15.1. Product Introduction

## 15.1.1. What is DTS?

Data Transmission Service (DTS) is a data service that is provided by Alibaba Cloud. DTS supports data transmission between various types of data sources, such as relational databases and big data systems.

### Features

DTS has the following advantages over traditional data migration and synchronization tools: high compatibility, high performance, security, reliability, and ease of use. DTS helps you simplify data transmission and focus on business development.

| Feature | Description |
| --- | --- |
| Data migration | You can use DTS to migrate data between homogeneous and heterogeneous data sources. This feature is suitable for the following scenarios: data migration to Alibaba Cloud, data migration between instances within Alibaba Cloud, and database splitting and scale-out. |
| Data synchronization | You can use DTS to synchronize data between data sources. This feature is suitable for the following scenarios: disaster recovery, data backup, load balancing, cloud BI systems, and real-time data warehousing. |

## 15.1.2. Benefits

DTS supports transmitting data between data sources such as relational databases and online analytical processing (OLAP) databases. DTS supports multiple data transmission modes, including data migration, data synchronization, and change tracking. Compared with other third-party data migration and synchronization tools, DTS supports diverse data transmission scenarios and ensures high performance, security, and reliability. DTS allows you to create and manage tasks with high efficiency.

### Multiple transmission modes

DTS supports multiple data transmission modes, including data migration, data synchronization, and change tracking. In the change tracking and data synchronization modes, data is transmitted in real time.

In the data migration mode, data is migrated between databases without interrupting application operations. The application downtime during data migration is reduced to a few minutes.

### High performance

DTS uses servers with high specifications to ensure the performance of each data synchronization or migration instance.

At the underlying layer of DTS, multiple measures are taken to improve performance.

Compared with traditional data synchronization tools, DTS provides better synchronization performance. You can use DTS to concurrently synchronize transactions and the incremental data of a single table.

DTS supports concurrent data capture and data write. The data capture and data write modules are decoupled to improve the overall throughput.

To implement concurrent migration between tables and within tables, DTS applies a variety of intelligent sharding policies on the tables to be migrated. This ensures parallel data synchronization and improves the processing efficiency of a large number of databases and a large amount of data in a single table.

DTS uses efficient network protocols and data compression technology to reduce the amount of data on long-distance connections and minimize data transmission latency. Multiple TCP connections are deployed to improve the synchronization performance and stability in abnormal network environments.

DTS provides horizontal scaling capabilities. You can improve capacity by increasing the number of tasks.

## High security and reliability

DTS is deployed based on clusters. If a node in a cluster is down or faulty, the control center moves all tasks from this node to another healthy node in the cluster within seconds.

DTS provides a 24 x 7 mechanism for validating data accuracy in some instances to discover and rectify inaccurate data. This helps ensure data integrity.

Secure transmission protocols and tokens are used for authentication across DTS modules to ensure reliable data transmission.

You can configure data verification tasks for DTS tasks to verify the data consistency between the source and destination databases. You can also flexibly adjust the verification parameters, such as the data volume of batch queries. This way, you can limit the bandwidth of verification tasks, control the I/O overhead of verification tasks, and minimize the impact on your business.

DTS supports the deployment of multiple data centers in the same city or across cities. If one of the data centers fails, you can quickly switch your workloads to another data center. In addition, if you create DTS tasks in different regions, you can centrally manage these tasks in the DTS console.

## Ease of use

The DTS console is a graphical user interface (GUI) that provides a wizard-like process to assist you in creating tasks for various scenarios.

To facilitate task management, the DTS console displays task information such as status, progress, and performance.

DTS supports resumable transmission, and monitors task status on a regular basis. If DTS detects a network failure or system error, DTS automatically fixes the failure or error and restarts the task. If the failure or error persists, you can manually troubleshoot and restart the task in the DTS console.

# 15.1.3. Environment requirements

Use Data Transmission Service (DTS) on hosts of the following models:

- PF51. *
- PV52P2M1. *
- DTS_E. *
- PF61. *
- PF61P1. *
- PV62P2M1. *
- PV52P1. *
- Q5F53M1. *
- PF52M2. *
- Q41. *
- Q5N1.22
- Q5N1.2B
- Q46.22
- Q46.2B
- W41.22
- W41.2B
- W1.22
- W1.2B
- W1.2C
- D13.12

Use the following operating system:

AliOS7U2-x86-64

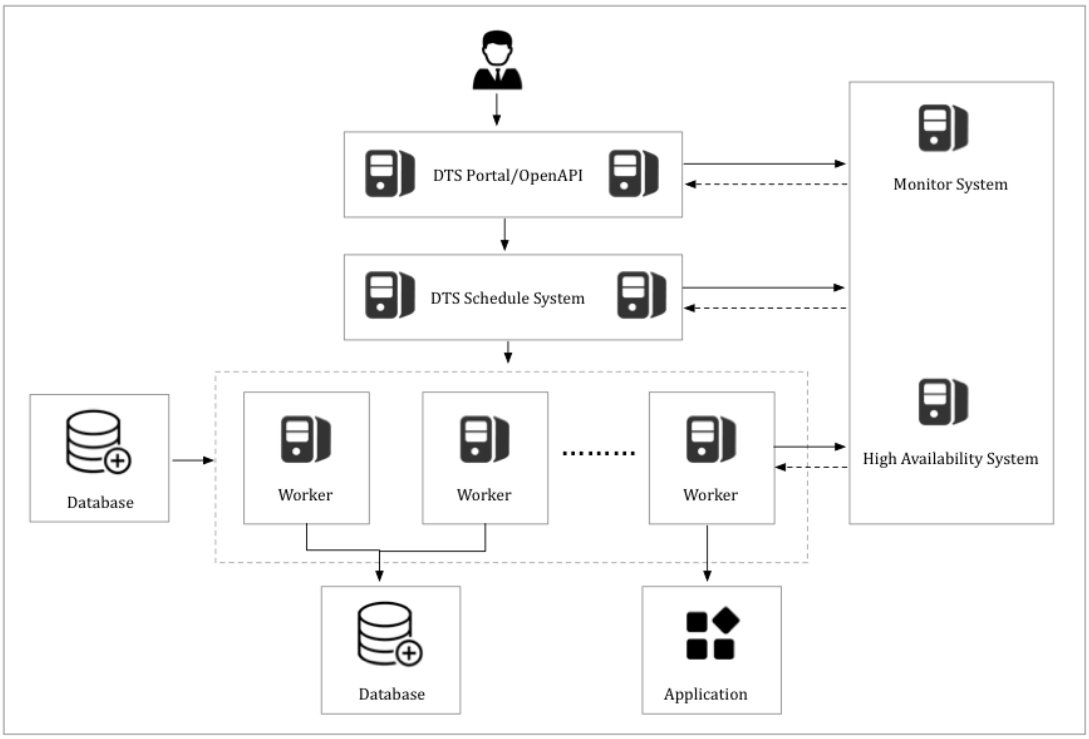
> **Important**
>
> - Do not use DTS on hosts whose models are excluded from the preceding list.
> - The */apsara* directory used by DTS resides on only one hard disk. Make sure that the space of the hard disk is larger than 2 TB.
>
>   If the space of the hard disk where the */apsara* directory resides is smaller than 2 TB, tasks may fail to run and errors may occur. In this case, DTS cannot restore failed tasks or pull data properly.

# 15.1.4. Architecture

## System architecture

System architecture shows the system architecture of DTS.

System architecture




- **High availability**

  Each DTS module comes with a primary-secondary architecture to ensure high availability of the system. The disaster recovery module runs a health check on each node in real time. Once a node exception is detected, the module switches the channel to another healthy node within seconds.

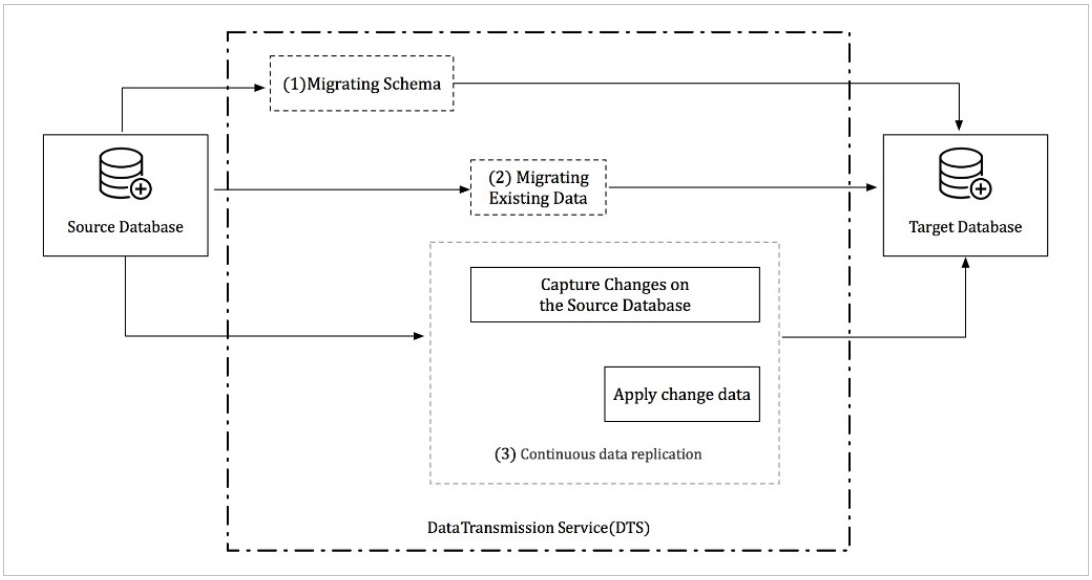
- **Monitor changes in the data source IP address**

  For data subscription and synchronization channels, the disaster recovery module checks for any changes. For example, once it detects a change in the data source address, the module dynamically changes the method for connecting to the data source to ensure channel stability.

## Data migration process

Data migration workflow shows how data migration works.

Data migration workflow




Data migration supports schema migration, real-time full data migration, and real-time incremental data migration. To implement migration without service interruption, follow these steps:

1. Schema migration
2. Full data migration
3. Incremental data migration

For migration between heterogeneous databases, DTS reads the schema using the syntax of the source database, translates the schema into the syntax of the destination database, and then imports the schema to the destination instance.

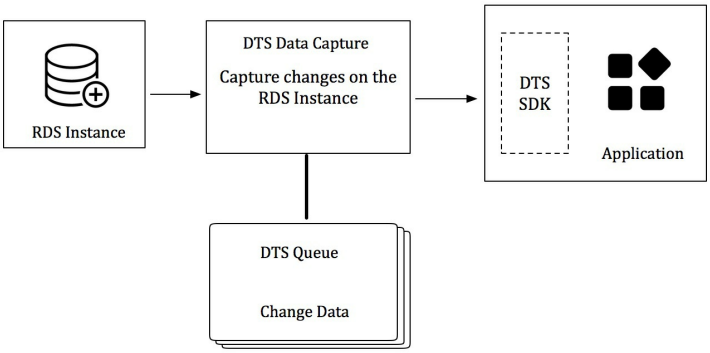
Full data migration takes a longer time. In this process, new data is continuously written into the source instance. To ensure data consistency, DTS starts the incremental data pulling module before full data migration. This module pulls the incremental data from the source instance and then parse, encapsulate, and store the data locally.

When full data migration is complete, DTS starts the incremental data playback module. The module retrieves the incremental data from the incremental data pulling module. After reverse parsing, filtering, and encapsulation, the data is synchronized to the destination instance. Eventually, data is synchronized between the source and destination instances in real time.

## Data subscription process

Data subscription workflow shows how data subscription works.

Data subscription workflow




Data subscription supports pulling incremental data from the RDS instance in real time. You can subscribe to the incremental data on the data subscription server using the DTS SDK. You can also customize data consumption based on business requirements.

The data pulling module of the DTS server captures raw data from the data source, and makes the incremental data locally persistent by parsing, filtering, and formatting it.

The data capturing module connects to the source instance using the database protocol and pulls the incremental data from the source instance in real time. For example, the data capturing module connects to an RDS for MySQL instance using the binlog dump command.

DTS guarantees the high availability of the data pulling module and downstream consumption SDKs.

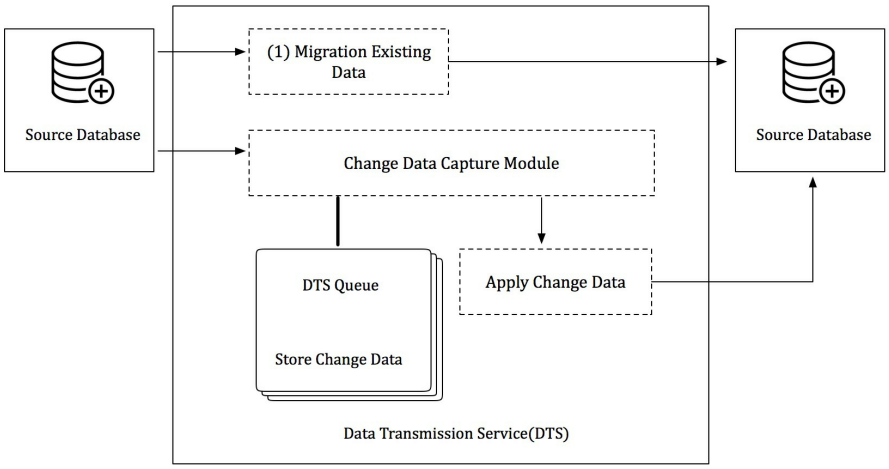
To ensure the high availability of the data pulling module, the DTS disaster recovery module restarts the data pulling module on a healthy service node once an exception is detected in the data pulling module.

The DTS server ensures the high availability of downstream consumption SDKs. If you start multiple consumption SDKs for the same subscription channel, the server pushes the incremental data to only one SDK at a time. If the consumption encounters an exception, the service end selects another consumption process from other healthy downstream nodes to push data to that consumption process. In this way, the high availability of downstream consumption processes can be guaranteed.

## Real-time synchronization workflow

Real-time synchronization workflow shows how real-time synchronization works.

Real-time synchronization workflow




The data synchronization feature in DTS enables real-time synchronization of incremental data between any two RDS instances.

To create a synchronization channel, follow these steps:

- Initial synchronization: The existing data in the source instance is synchronized to the destination instance.
- Incremental data synchronization: After initial synchronization, the incremental data starts to be synchronized between the source instance and destination instance in real time. During this phase, data is eventually synchronized between the source and destination instances.

DTS provides the following underlying modules for real-time incremental data synchronization:

- Data reading module

  The data reading module reads raw data from the source instance and makes the data locally persistent by parsing, filtering, and formatting it. The data reading module connects to the source instance using the database protocol and reads the incremental data from the source instance. For example, the data reading module connects to an RDS for MySQL instance using the binlog dump command.

- Data playback module

  The data playback module requests incremental data from the data reading module, filters data based on the objects to be synchronized, and then synchronize the data to the destination instance without compromising the transaction sequence and consistency.

DTS ensures the high availability of the data reading module and data playback module. When a channel exception is detected, the disaster recovery module restarts the channel on a healthy service node. In this way, the high availability of the synchronization channels is guaranteed.

# 15.1.5. Features

## 15.1.5.1. Data migration

You can use Data Transmission Service (DTS) to migrate data between various types of data sources. Typical scenarios include data migration to the cloud, data migration between instances within Apsara Stack, and database splitting and scale-out. DTS supports data migration between homogeneous and heterogeneous data sources. DTS provides the following extract, transform, and load (ETL) features: object name mapping and data filtering.

## Supported databases

| Source database | Destination database | Migration type |
| --- | --- | --- |
| • Self-managed MySQL database<br>Version 5.1, 5.5, 5.6, 5.7, or 8.0<br>• ApsaraDB RDS for MySQL<br>All versions | Self-managed MySQL database<br>Version 5.1, 5.5, 5.6, 5.7, or 8.0 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | ApsaraDB RDS for MySQL<br>All versions | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | PolarDB-X (formerly known as DRDS)<br>All versions | • Full data migration<br>• Incremental data migration |
| | Self-managed Oracle database (RAC or PDB instance)<br>Version 9i, 10g, 11g, 12c, 18c, or 19c | • Full data migration<br>• Incremental data migration |
| | Self-managed Kafka cluster<br>Versions 0.1 to 2.0 | • Schema migration<br>• Full data migration<br>• Incremental data migration |

| Source database | Destination database | Migration type |
| --- | --- | --- |
| Self-managed SQL Server database<br>Version 2005, 2008, 2008 R2, 2012, 2014, 2016, or 2017<br>Note<br>• DTS does not support SQL Server clusters or SQL Server Always On availability groups (AOAGs).<br>• If the version of the source database is 2005, incremental data migration is not supported. | • Self-managed SQL Server database<br>Version 2005, 2008, 2008 R2, 2012, 2014, 2016, or 2017<br>Note DTS does not support SQL Server clusters or SQL Server Always On availability groups (AOAGs).<br>• ApsaraDB RDS for SQL Server<br>Version 2012, 2014, 2016, or 2017 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | Self-managed Oracle database (RAC or PDB instance)<br>Version 9i, 10g, 11g, 12c, 18c, or 19c | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | PolarDB<br>Version 9.3, 9.6, 10, or 11 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | Self-managed MySQL database<br>Version 5.1, 5.5, 5.6, 5.7, or 8.0 | • Schema migration<br>• Full data migration<br>• Incremental data migration |

| Source database | Destination database | Migration type |
| --- | --- | --- |
| Self-managed Oracle database (RAC or PDB instance)<br>Version 9i, 10g, 11g, 12c, 18c, or 19c | ApsaraDB RDS for MySQL<br>All versions | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | PolarDB-X<br>All versions | • Full data migration<br>• Incremental data migration |
| | AnalyticDB for MySQL<br>Version 2.0 or 3.0 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | AnalyticDB for PostgreSQL (formerly known as HybridDB for PostgreSQL)<br>Version 4.3 or 6.0 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | Self-managed PostgreSQL database<br>Version 9.4, 9.5, 9.6, 10.x, 11, or 12 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| • Self-managed PostgreSQL database<br>Version 9.4, 9.5, 9.6, 10.x, 11, or 12<br>• ApsaraDB RDS for PostgreSQL<br>Version 9.4 or 10 | • Self-managed PostgreSQL database<br>Version 9.4, 9.5, 9.6, 10.x, 11, or 12<br>• ApsaraDB RDS for PostgreSQL<br>Version 9.4 or 10 | • Schema migration<br>• Full data migration<br>• Incremental data migration |

| Source database | Destination database | Migration type |
| --- | --- | --- |
| PolarDB<br>Version 9.3, 9.6, 10, or 11 | Self-managed Kafka cluster<br>Versions 0.1 to 2.0 | Incremental data migration |
| | Self-managed Oracle database (RAC or PDB instance)<br>Version 9i, 10g, 11g, 12c, 18c, or 19c | • Full data migration<br>• Incremental data migration |
| | PolarDB<br>Version 9.3, 9.6, 10, or 11 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| Self-managed Redis database<br>Version 2.8, 3.0, 3.2, 4.0, or 5.0 | Self-managed Redis database<br>Version 2.8, 3.0, 3.2, 4.0, or 5.0 | • Full data migration<br>• Incremental data migration |
| Self-managed MongoDB database<br>Version 3.0, 3.2, 3.4, 3.6, 4.0, or 4.2 | Self-managed MongoDB database<br>Version 3.0, 3.2, 3.4, 3.6, 4.0, or 4.2 | • Full data migration<br>• Incremental data migration |
| MaxCompute<br>All versions | • Self-managed MySQL database<br>Version 5.1, 5.5, 5.6, 5.7, or 8.0<br>• ApsaraDB RDS for MySQL<br>All versions | • Schema migration<br>• Full data migration |
| PolarDB-X 2.0 | • PolarDB-X 2.0<br>Note The PolarDB-X 2.0 instance must be compatible with MySQL V5.7.<br>• AnalyticDB for PostgreSQL | • Schema migration<br>• Full data migration<br>• Incremental data migration |

| Source database | Destination database | Migration type |
| --- | --- | --- |
| • Self-managed MySQL database<br>Versions 5.1, 5.5, 5.6, 5.7, and 8.0<br>• ApsaraDB RDS for MySQL<br>All versions | Self-managed MySQL database<br>Versions 5.1, 5.5, 5.6, 5.7, and 8.0 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | ApsaraDB RDS for MySQL<br>All versions | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | PolarDB-X (formerly known as DRDS)<br>All versions | • Full data migration<br>• Incremental data migration |
| | Self-managed Oracle database in a RAC or non-RAC architecture<br>Versions 9i, 10g, 11g, 12c, 18c, and 19c | • Full data migration<br>• Incremental data migration |
| | Self-managed Kafka cluster<br>Versions 0.1 to 2.0 | • Schema migration<br>• Full data migration<br>• Incremental data migration |

| Source database | Destination database | Migration type |
| --- | --- | --- |
| Self-managed SQL Server database<br>Versions 2005, 2008, 2008 R2, 2012, 2014, 2016, and 2017<br>Note<br>• DTS does not support SQL Server clusters or SQL Server Always On availability groups (AOAGs).<br>• If the version of the source database is 2005, incremental data migration is not supported. | • Self-managed SQL Server database<br>Versions 2005, 2008, 2008 R2, 2012, 2014, 2016, and 2017<br>Note DTS does not support SQL Server clusters or SQL Server AOAGs.<br>• ApsaraDB RDS for SQL Server<br>Versions 2012, 2014, 2016, and 2017 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | Self-managed Oracle database in a RAC or non-RAC architecture<br>Versions 9i, 10g, 11g, 12c, 18c, and 19c | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | PolarDB<br>Versions 9.3, 9.6, 10, and 11 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | Self-managed MySQL database<br>Versions 5.1, 5.5, 5.6, 5.7, and 8.0 | • Schema migration<br>• Full data migration<br>• Incremental data migration |

| Source database | Destination database | Migration type |
| --- | --- | --- |
| Self-managed Oracle database in a RAC or non-RAC architecture<br>Versions 9i, 10g, 11g, 12c, 18c, and 19c | ApsaraDB RDS for MySQL<br>All versions | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | PolarDB-X<br>All versions | • Full data migration<br>• Incremental data migration |
| | AnalyticDB for MySQL<br>Versions 2.0 and 3.0 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | AnalyticDB for PostgreSQL (formerly known as HybridDB for PostgreSQL)<br>Versions 4.3 and 6.0 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| | Self-managed PostgreSQL database<br>Versions 9.4, 9.5, 9.6, 10.x, 11, and 12 | • Schema migration<br>• Full data migration<br>• Incremental data migration |

| Source database | Destination database | Migration type |
| --- | --- | --- |
| • Self-managed PostgreSQL database<br>Versions 9.4, 9.5, 9.6, 10.x, 11, and 12<br>• ApsaraDB RDS for PostgreSQL<br>Versions 9.4 and 10 | • Self-managed PostgreSQL database<br>Versions 9.4, 9.5, 9.6, 10.x, 11, and 12<br>• ApsaraDB RDS for PostgreSQL<br>Versions 9.4 and 10 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| PolarDB<br>Versions 9.3, 9.6, 10, and 11 | Self-managed Kafka cluster<br>Versions 0.1 to 2.0 | Incremental data migration |
| | Self-managed Oracle database in a RAC or non-RAC architecture<br>Versions 9i, 10g, 11g, 12c, 18c, and 19c | • Full data migration<br>• Incremental data migration |
| | PolarDB<br>Versions 9.3, 9.6, 10, and 11 | • Schema migration<br>• Full data migration<br>• Incremental data migration |
| Self-managed Redis database<br>Versions 2.8, 3.0, 3.2, 4.0, and 5.0 | Self-managed Redis database<br>Versions 2.8, 3.0, 3.2, 4.0, and 5.0 | • Full data migration<br>• Incremental data migration |
| Self-managed MongoDB database<br>Versions 3.0, 3.2, 3.4, 3.6, 4.0 and 4.2 | Self-managed MongoDB database<br>Versions 3.0, 3.2, 3.4, 3.6, 4.0 and 4.2 | • Full data migration<br>• Incremental data migration |
| MaxCompute<br>All versions | • Self-managed MySQL database<br>Versions 5.1, 5.5, 5.6, 5.7, and 8.0<br>• ApsaraDB RDS for MySQL<br>All versions | • Schema migration<br>• Full data migration |

| Source database | Destination database | Migration type |
| --- | --- | --- |
| PolarDB-X 2.0 | • PolarDB-X 2.0<br>Note Make sure that your PolarDB-X 2.0 instances are compatible with MySQL 5.7.<br>• AnalyticDB for PostgreSQL | • Schema migration<br>• Full data migration<br>• Incremental data migration |

## Online migration

DTS uses online migration. You must configure the source instance, the destination instance, and the objects to be migrated. DTS automatically completes the entire data migration process. You can select all of the supported migration types to minimize the impact of online data migration on your services. However, you must ensure that DTS servers can connect to both the source and destination instances.

## Data migration types

DTS supports schema migration, full data migration, and incremental data migration.

- Schema migration: DTS migrates schemas from the source instance to the destination instance.
- Full data migration: migrates historical data from the source instance to the destination instance.
- Incremental data migration: DTS synchronizes incremental data that is generated during data migration from the source instance to the destination instance. You can select schema migration, full data migration, and incremental migration to migrate data with minimal downtime.

## ETL features

Data migration supports the following ETL features:

- Object name mapping: You can rename the columns, tables, and databases that are migrated to the destination database.
- Data filtering: You can use SQL conditions to filter the required data in a specific table. For example, you can specify a time range to migrate only the latest data.

## Alerts

If an error occurs during data migration, DTS immediately sends a text message to the task owner. This allows the owner to handle the error at the earliest opportunity.

## Migration task

A migration task is a basic unit of data migration. To migrate data, you must create a migration task in the DTS console. To create a migration task, you must configure the required information such as the source and destination instances, migration types, and objects to be migrated. You can create, manage, stop, and delete migration tasks in the DTS console.

The table Task statuses describes the statuses of a migration task.

### Task statuses

| Task status | Description | Available operation |
| --- | --- | --- |
| Not Started | The migration task is configured but no precheck is performed. | • Run a precheck<br>• Delete the migration task |
| Prechecking | A precheck is being performed but the migration task is not started. | Delete the migration task |
| Passed | The migration task has passed the precheck but has not been started. | • Start the migration task<br>• Delete the migration task |
| Migrating | The task is migrating data. | • Pause the migration task<br>• Stop the migration task<br>• Delete the migration task |
| Migration Failed | An error occurred during data migration. You can identify the point of failure based on the progress of the migration task. | Delete the migration task |
| Paused | The migration task is paused. | • Start the migration task<br>• Delete the migration task |
| Completed | The migration task is completed, or you have stopped data migration by clicking **End**. | Delete the migration task |

# 15.1.5.2. Data synchronization

You can use Data Transmission Service (DTS) to synchronize data between two data sources. This feature is suitable for various scenarios, such as data backup, disaster recovery, active geo-redundancy, cross-border data synchronization, load balancing, cloud BI systems, and real-time data warehousing.

## Supported databases

## Objects that can be synchronized

- You can select columns, tables, or databases as the objects to be synchronized. You can specify one or more tables that you want to synchronize.
- DTS allows you to synchronize data between tables that have different names, or between databases that have different names. You can use the object name mapping feature to specify the names of destination columns, tables, and databases.
- You can specify one or more columns that you want to synchronize.

## Synchronization tasks

A synchronization task is a basic unit of data synchronization. To synchronize data between two instances, you must create a synchronization task in the DTS console.

The following table describes the statuses of a synchronization task.

Task statuses

| Task status | Description | Available operation |
| --- | --- | --- |
| Prechecking | A precheck is being performed before the synchronization task is started. | • View the configurations of the synchronization task<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure an alert rule for the synchronization task |

| Task status | Description | Available operation |
| --- | --- | --- |
| Precheck Failed | The synchronization task has failed to pass the precheck. | • Run a precheck<br>• View the configurations of the synchronization task<br>• Reselect the objects to be synchronized<br>• Modify the synchronization speed<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure an alert rule for the synchronization task |
| Not Started | The synchronization task has passed the precheck but has not been started. | • Run a precheck<br>• Start the synchronization task<br>• Reselect the objects to be synchronized<br>• Modify the synchronization speed<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure an alert rule for the synchronization task |
| Performing Initial Synchronization | Initial synchronization is being performed. | • View the configurations of the synchronization task<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure an alert rule for the synchronization task |

| Task status | Description | Available operation |
| --- | --- | --- |
| Initial Synchronization Failed | The task has failed during initial synchronization. | • View the configurations of the synchronization task<br>• Reselect the objects to be synchronized<br>• Modify the synchronization speed<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure an alert rule for the synchronization task |
| Synchronizing | The task is synchronizing data. | • View the configurations of the synchronization task<br>• Reselect the objects to be synchronized<br>• Modify the synchronization speed<br>• Pause the synchronization task<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure an alert rule for the synchronization task |
| Synchronization Failed | An error has occurred during synchronization. | • View the configurations of the synchronization task<br>• Reselect the objects to be synchronized<br>• Modify the synchronization speed<br>• Start the synchronization task<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure an alert rule for the synchronization task |

| Task status | Description | Available operation |
| --- | --- | --- |
| Paused | The synchronization task is paused. | • View the configurations of the synchronization task<br>• Reselect the objects to be synchronized<br>• Modify the synchronization speed<br>• Start the synchronization task<br>• Delete the synchronization task<br>• Replicate the configurations of the synchronization task<br>• Configure an alert rule for the synchronization task |

## Advanced features

You can use the following advanced features to facilitate data synchronization:

- Add or remove the objects to be synchronized

  You can add or remove the required objects when a task is synchronizing data.

- View and analyze the synchronization performance

  DTS provides trend charts that allow you to view and analyze the performance of your synchronization tasks. The synchronization performance is measured based on bandwidth, synchronization speed (TPS), and synchronization delay.

- Monitor synchronization tasks

  DTS allows you to monitor the status of synchronization tasks. If the threshold for synchronization delay is reached, you will receive an alert notification. You can set the alert threshold based on the sensitivity of your businesses to synchronization delays.

# 15.1.5.3. Change tracking

You can use Data Transmission Service (DTS) to track data changes from databases in real time. This feature applies to the following scenarios: cache updates, business decoupling, asynchronous data processing, synchronization of heterogeneous data, and synchronization of extract, transform, and load (ETL) operations.

## Supported databases

- User-created MySQL database or ApsaraDB RDS for MySQL instance
- Cloud Native Distributed Database PolarDB-X (formerly known as DRDS)
- User-created Oracle database

## Objects for change tracking

The objects for change tracking include tables and databases. You can specify one or more tables from which you want to track data changes.

In change tracking, data changes include data manipulation language (DML) operations and data definition language (DDL) operations. When you configure a change tracking task, you can select the operation type.

## Change tracking channel

A change tracking channel is the basic unit of change tracking and data consumption. To track data changes from an RDS instance, you must create a change tracking channel for the RDS instance in the DTS console. The change tracking channel pulls incremental data from the RDS instance in real time and locally stores the incremental data. You can use the DTS SDK to consume the incremental data from the change tracking channel. You can also create, manage, or delete change tracking channels in the DTS console.

A change tracking channel can be consumed by only one downstream SDK client. To track data changes from an RDS instance by using multiple downstream SDK clients, you must create an equivalent number of change tracking channels. The channels pull incremental data from the same RDS instance.

The following table describes the statuses of a change tracking channel.

Channel statuses

| Channel status | Description | Available operation |
| --- | --- | --- |
| Prechecking | The configuration of the change tracking channel is complete and a precheck is being performed. | Delete the change tracking channel |
| Not Started | The change tracking channel has passed the precheck but has not been started. | • Start the change tracking channel<br>• Delete the change tracking channel |
| Performing Initial Change Tracking | The initial change tracking is in progress. This process takes about 1 minute. | Delete the change tracking channel |
| Normal | Incremental data is being pulled from the source RDS instance. | • View the demo code<br>• View the tracked data<br>• Delete the change tracking channel |
| Error | An error occurs when the change tracking channel pulls incremental data from the source RDS instance. | • View the demo code<br>• Delete the change tracking channel |

## Advanced features

You can use the following advanced features that are provided for change tracking:

- Add or remove the objects for change tracking

You can add or remove the required objects when a change tracking task is running.

- View the tracked data

  You can view the data that is tracked from the change tracking channel in the DTS console.

- Modify consumption checkpoints

  You can modify consumption checkpoints.

- Monitor change tracking channels

  DTS allows you to monitor the status of change tracking channels. If the threshold for consumption delay is reached, you will receive an alert. You can set the alert threshold based on the sensitivity of your businesses to consumption delays.

# 15.1.5.4. ETL

DTS provides the extract, transform, and load (ETL) feature. When you configure a migration or synchronization task, you can rename databases, tables, and columns in the destination instance and set conditions to filter specific data.

## ETL

You can perform drag-and-drop operations to configure ETL tasks. You can use the ETL feature to clean and transform streaming data, and to accurately and efficiently obtain the data that you need.

You can add transformation components between the source and destination databases. You can transform data and write the processed data to the destination database in real time. For example, you can join a stream table and a dimension table into a large table and write the data of the large table to the destination database for query and analysis. You can also add a field to the source table and configure a function to assign values to the field. Then, you can write the field to the destination database.

- Supported databases

| Source or destination database | Supported database |
| --- | --- |
| Source database | Self-managed MySQL database |
| Destination database | Self-managed MySQL database |

- Supported transformation components
  - JOIN: allows you to join two tables into one table.
  - Field Calculator: supports more than 90 function compute scenarios to transform data.
  - Table Record Filter: allows you to filter data by using WHERE conditions.
- Scenarios
  - Centralized management of multi-region or heterogeneous data in real time: To facilitate centralized and efficient management and decision-making, you can store heterogeneous data or data from different regions to the same database in real time.

- Real-time data integration: The data processing capabilities of ETL greatly improves the efficiency of data integration. The low-code development mode reduces the difficulty and cost of data integration.
- Real-time data warehousing: The ETL feature provides industry-leading streaming data processing capabilities to help you quickly build real-time data warehouses.
- Acceleration of on-premises data warehouses: In streaming data processing, pre-processed data is shipped to data warehouses for in-depth mining. The data warehouses can provide services without affecting your business systems.
- Real-time reporting: To improve the efficiency of reporting and facilitate digital transformation, you can build a real-time reporting system. The system is suitable for various real-time analysis scenarios.
- Real-time computing: You can clean the streaming data generated on the business side in real time to extract feature values and tags. Typical scenarios include online business computing models (such as profiling, risk control, and recommendations) and real-time big screens.

- **Advanced features**
  - Industry-leading computing effectiveness: The ETL feature is integrated with the data collection capabilities of DTS. The ETL feature ensures data accuracy and has industry-leading computing effectiveness.
  - Flexible task monitoring and management: You can monitor and manage ETL tasks in the DTS console. For example, you can start a task, stop a task, and view task details.

### Object name mapping

When you configure a data migration or synchronization task, you can rename objects such as databases, tables, and columns, and map the object names to the names required by the destination instance.

DTS allows you to rename multiple tables and columns at a time.

### Filter data

You can specify filter conditions to exclude databases, tables, and columns from synchronization. You can also write WHERE conditions to filter row records and migrate or synchronize only the data that meets the conditions to the destination table.

## 15.1.5.5. Data consistency

Data Transmission Service (DTS) uses the data reading module, data loading module, data verification feature, and resumable transmission feature to ensure data consistency between the source and destination databases.

### Data reading module and data loading module

DTS uses the data reading module and data loading module to ensure the consistency of incremental data.

Data reading module: The data reading module obtains raw data from the source instance. After parsing, filtering, and syntax conversion, the data is persisted locally in the DTS server.

> **Note** The data reading module connects to the source instance over the corresponding protocol and reads the incremental data of the source instance. For example, the data reading module connects to an ApsaraDB RDS for MySQL instance over the binlog dump protocol.

Data loading module: The data loading module retrieves incremental data from the data reading module and filters the data based on the configured synchronization objects. The data loading module then synchronizes the data to the destination instance without compromising transactional sequence and consistency.

### Data verification

After data migration or data synchronization is completed, DTS performs data verification in the O&M platform to ensure the integrity and consistency of data that is replicated to the destination instance.

DTS detects the number of inconsistent data records between the source and destination instances. You can click the number displayed in the Diff Row column to view details about the verified objects and the inconsistent data records. Then, you can revise data based on the provided information.

### Resumable transmission

During data migration and data synchronization, DTS records the timestamp when data records are replicated to the destination instance. If a network failure occurs, DTS records the timestamp of the data records that have been replicated to the destination instance when the failure occurs. When the failure is recovered, DTS resumes data migration or data synchronization from the timestamp before the failure occurs. The resumable transmission feature ensures data consistency.

## 15.1.6. Scenarios

DTS supports multiple features including data migration, real-time data subscription, and real-time data synchronization to meet the following scenarios.

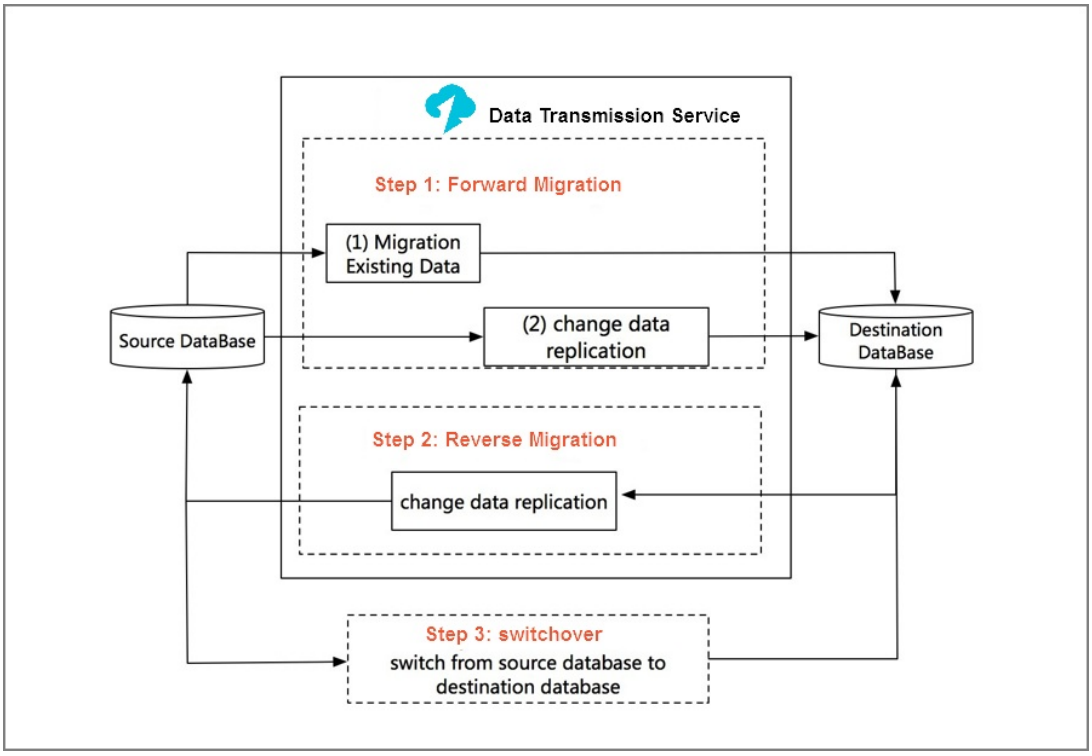
### Migration with service downtime reduced to minutes

Many users seek for a way to migrate systems without affecting their services. However, data changes if services are not suspended during the migration. To ensure data consistency, many third-party migration tools require that the service be suspended during data migration. It may take hours or even days throughout the migration and result in a significant loss in service availability.

To reduce the barrier of database migration, DTS provides an interruption-free migration solution that minimizes the service downtime to minutes.

Interruption-free migration shows how interruption-free migration works.

Interruption-free migration



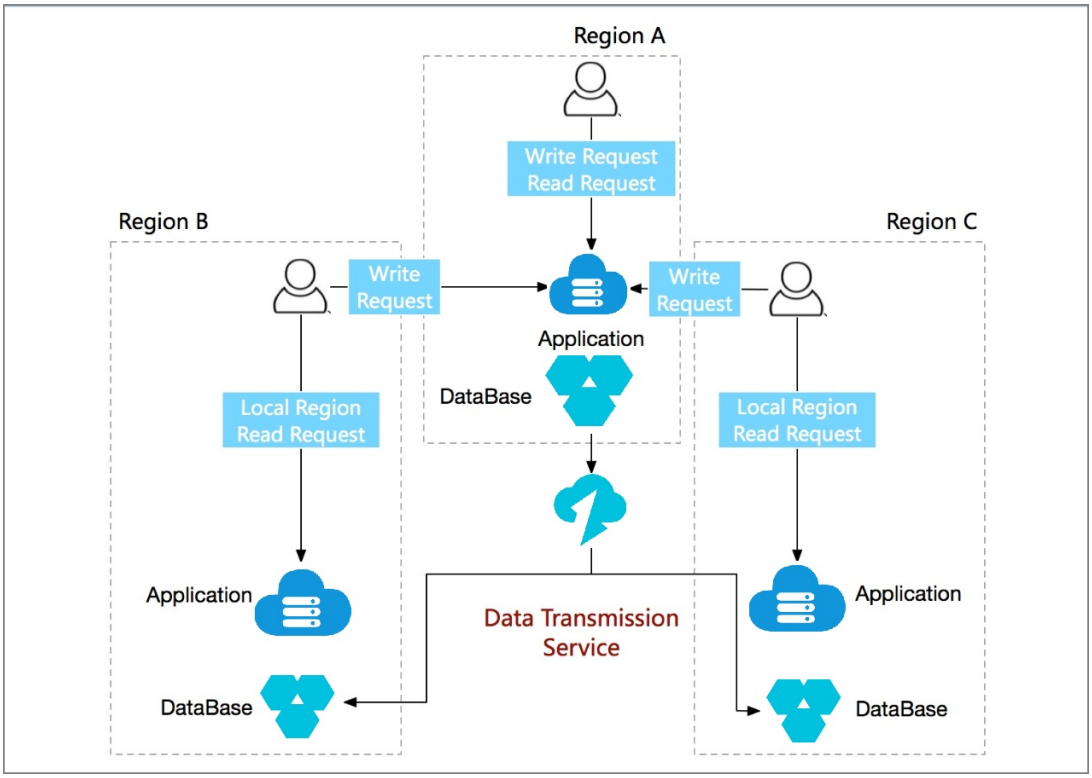

The interruption-free data migration process involves schema migration, full data migration, and incremental data migration. In the incremental data migration phase, data is synchronized between the source and destination instances in real time. You can validate the service in the destination database. After the validation is complete, the service is migrated to the destination database. The entire system is then eventually migrated.

Throughout the migration process, the service experiences interruptions only when it is switched from the source instance to the destination instance.

## Accelerated access to global services to empower cross-border businesses

If services with widely distributed users, such as global services, are deployed only in one region, users in other regions have to access them remotely, resulting in high access latency and poor user experience. To accelerate the access to global services and improve access experience, you can adjust the architecture, as shown in Reduced cross-region access latency.

Reduced cross-region access latency



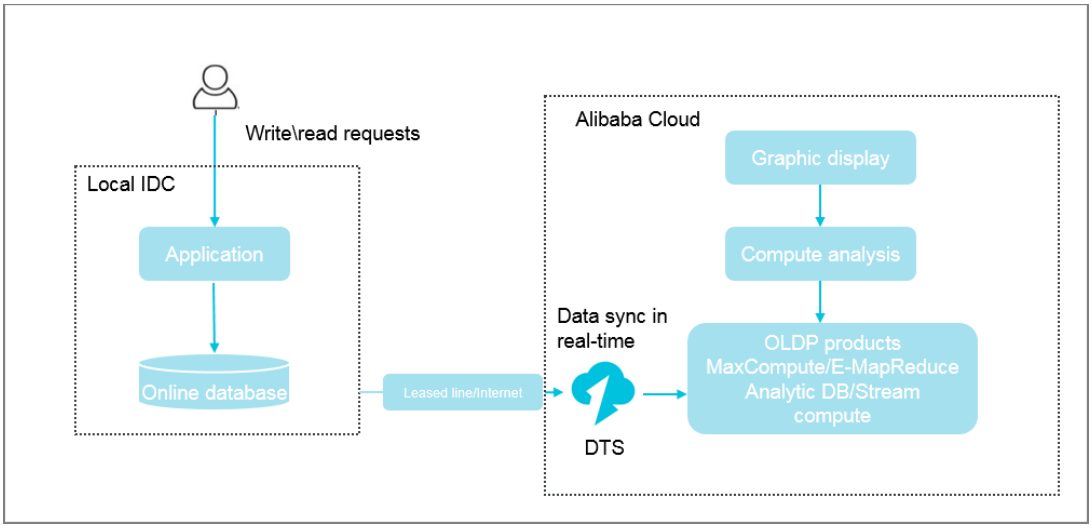

This architecture consists of one center and multiple units. Write requests of users in all regions are routed back to the center. DTS synchronizes data in the center to all units. Read requests of users in different regions can be routed to nearby units to avoid remote access and reduce access latency. In this way, access to global services is accelerated.

## Custom cloud BI system built with more efficiency

User-created business intelligence (BI) systems cannot meet the increasing demand for real-time performance and are difficult to manipulate. With the Apsara Stack BI architecture, you can quickly build a BI system without affecting the current architecture. For this reason, more and more users choose to build BI systems that meet their own business requirements on Apsara Stack.

DTS can help you synchronize data stored in local databases to an Apsara Stack BI system (such as MaxCompute or StreamCompute) in real time. You can then perform subsequent data analysis with various compute engines while viewing the computing results in real time with a visualization tool. You can also synchronize those results back to the local IDC with a migration tool. Cloud BI architecture shows the implementation architecture.

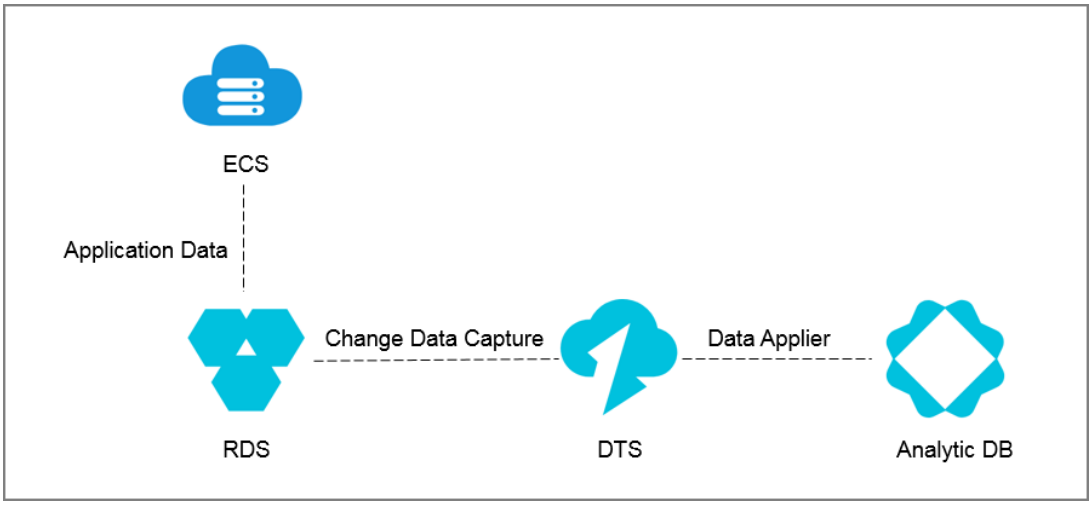
Cloud BI architecture




## Real-time data analysis to rapidly respond to market conditions

Data analysis is essential in improving enterprise insights and user experience. Real-time data analysis enables enterprises to adjust marketing strategies more quickly and flexibly so that they can adapt to the rapidly changing marketing conditions and demands for higher user experience. To implement real-time data analysis without affecting online services, service data needs to be synchronized to the analysis system in real time. For this reason, acquiring service data in real time becomes essential. In DTS, the data subscription feature can help you acquire real-time incremental data without affecting online services and synchronize the data to the analysis system using the SDK for real-time data analysis, as shown in Real-time data analysis.

Real-time data analysis

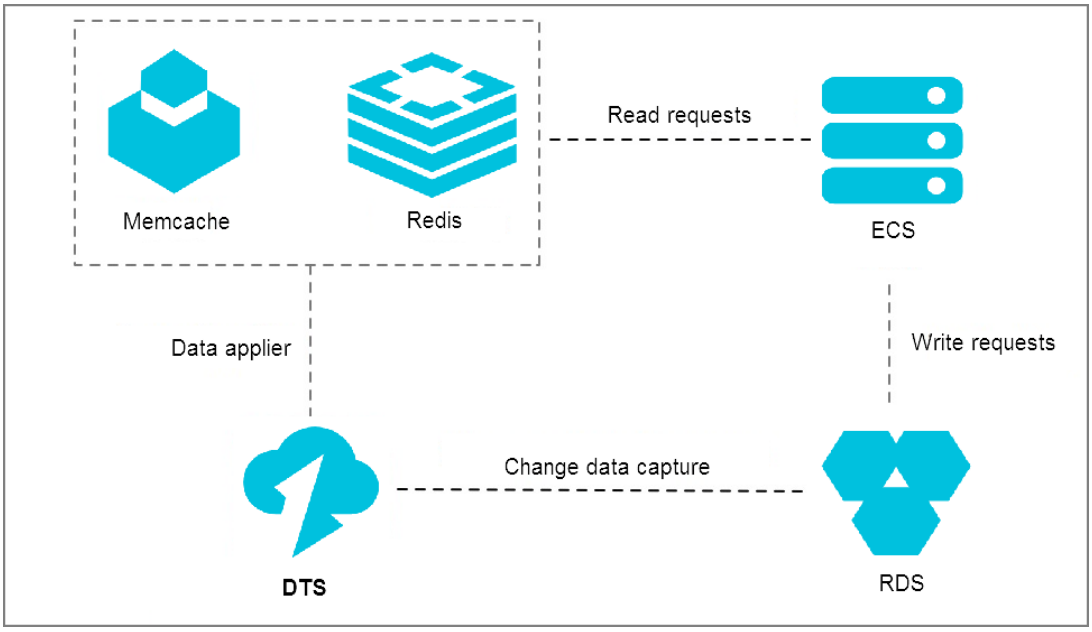



## Lightweight cache update policies to make core services more simple and reliable

To accelerate service access and improve concurrent read performance, many enterprises introduce the caching layer to the service architecture. In this architecture, all the read requests are routed to the caching layer, and the memory reading mechanism greatly improves read performance. Cached data cannot persist. If caching ends abnormally, data in the cache memory is lost. To ensure data integrity, the updated service data is kept in a persistent storage medium, such as a database.

In this condition, the service data is inconsistent between the cache and the persistent databases. The data subscription feature can help asynchronously subscribe to the incremental data in those databases and update the cached data to implement lightweight cache update policies. Cache update policies shows the architecture of these policies.

Cache update policies




Cache update policies offer the following benefits:

- Quick update with low latency

  Cache invalidation is an asynchronous process, and the service returns data directly after the database update is complete. For this reason, you do not need to consider the cache invalidation process, and the entire update path is short with low latency.

- Simple and reliable applications

  The complex doublewrite logic is not required for the application. You only need to start the asynchronous thread to monitor the incremental data and update the cached data.

- Application updates without extra performance consumption

  Because data subscription acquires incremental data by parsing incremental logs in the database, the acquisition process does not damage the performance of services and databases.

## Asynchronous service decoupling to make core services simpler and more reliable

Data subscription optimizes intensive coupling to asynchronous coupling by using real-time message notifications. This makes the core service logic simpler and more reliable. This application has been widely implemented in Alibaba. Tens of thousands of downstream services in the Taobao ordering system acquire real-time data updates through data subscription to trigger the business logic every day.

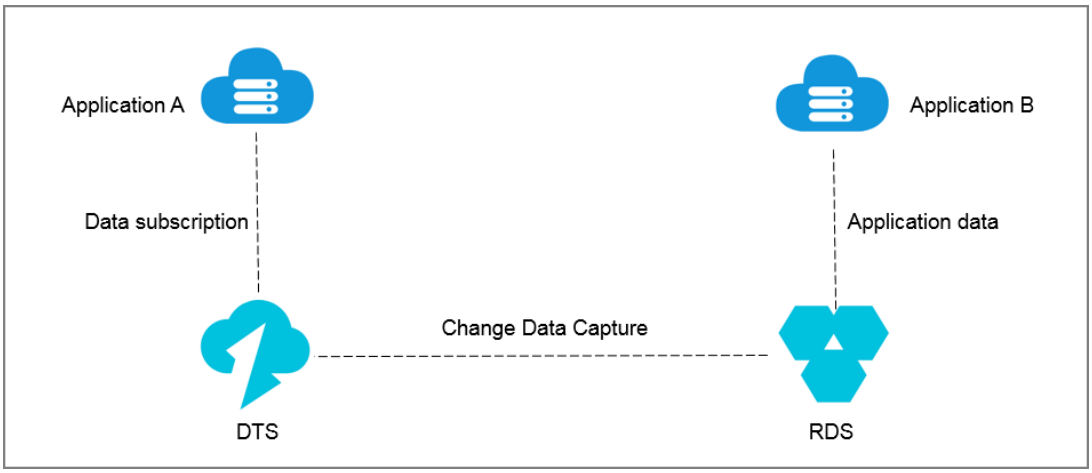
The following uses a simple example to describe the benefits of implementing data subscription in this scenario.

The e-commerce industry involves multiple services including the order management system, inventory management, and the shipping of goods. An ordering process with all of those services included is as follows: After a user places an order, downstream services including seller inventory notification and goods shipping are modified. When all logic modifications are complete, the order result is returned to the user. However, this ordering logic has the following issues:

- The lengthy ordering process results in poor user experience.
- The system is unstable and any downstream fault directly affects the availability of the ordering system.

To improve user experience of core applications, you can decouple the core applications and the dependent downstream services so that they can work asynchronously. In this way, the core applications become more stable and reliable. Asynchronous service decoupling shows how to adjust the logic.

Asynchronous service decoupling




The ordering system returns the order result directly after order placement. With DTS, the underlying layer acquires the updated data from the ordering system in real time. Then, the downstream service subscribes to the modified data using the SDK and triggers the service logic such as inventory and shipping. In this way, the ordering system becomes simpler and more reliable.

### Horizontal scaling to improve read performance and quickly adapt to business growth

A single RDS instance may not be able to support a large number of read requests, which may affect the main service process. To elastically improve the read performance and reduce database workload, you can create read-only instances using the real-time synchronization feature of DTS. These read-only instances take on large amounts of the database reading workload and expand the throughput of applications.

## 15.1.7. Concepts

### Precheck

Precheck is an essential stage before a migration task starts. It mainly checks the prerequisites that may affect a successful migration, such as the connectivity of the source and destination instances and the permissions of the migration accounts. If the precheck fails, you can fix the problems as instructed and run the precheck again.

## Schema migration

Schema migration is a type of migration tasks. In database migration, it refers to migrating the schema syntax, including tables, views, triggers, stored procedures, stored functions, and synonyms. For migration between heterogeneous databases, data types are mapped during schema migration, and the schema syntax is adjusted according to the schema syntax of the source and destination instances.

## Full data migration

Full data migration is a type of migration task. It refers to migrating all the data except the schema syntax from the source instance to the destination instance. If you select Full Data Migration only and leave Schema Migration unselected, new data generated in the source instance will not be migrated to the destination instance.

## Incremental data migration

Incremental data migration is a type of migration tasks. It refers to synchronizing the new data written to the source instance to the destination instance during the migration. When creating a migration task, if you select both Full Data Migration and Incremental Data Migration, DTS will first perform a static snapshot on the source instance, migrate the snapshot data to the destination instance, and then synchronize the new data from the source instance to the destination instance during the migration. Incremental data migration is a process of synchronizing data between the source and destination instances in real time. This process does not automatically end. If you want to stop migrating data, you must manually disable the task in the console.

## Initial synchronization

Initial synchronization refers to synchronizing the historical data of the objects to be synchronized to the destination instance before synchronizing the incremental data through the synchronization channel.

Initial synchronization includes initial schema synchronization and initial full data synchronization. Initial schema synchronization refers to synchronizing the required schema syntax in the initial stage. Initial full data synchronization refers to synchronizing the data of the objects for the first time.

## Synchronization performance

Synchronization performance is measured based on the number of records that are synchronized to the destination instance per second. The measurement unit is records per second (RPS).

## Synchronization delay

Synchronization delay refers to the duration between the timestamp when the latest data in the destination instance is starting to be synchronized from the source instance and the current timestamp of the source instance. It reflects the time difference between the data in the source and destination instances. If the synchronization delay is zero, data in the source instance is in sync with that in the destination instance.

## Subscription channel ID

The subscription channel ID is a unique identifier of a subscription channel. After you purchase a subscription channel, DTS automatically generates a subscription channel ID. To consume the incremental data using the SDK, you must configure a correct subscription channel ID. You can find the ID that corresponds to each subscription channel in the subscription list of the DTS console.

## Data update

In DTS, you can update data or its schema. A data update only modifies the data. The schema syntax is not changed. Operations including INSERT, UPDATE, and DELETE fall into this category.

## Schema update

In DTS, you can update data or its schema. Schema update modifies the schema syntax. Operations including CREATE TABLE, ALTER TABLE, and DROP VIEW fall into this category . You can choose whether to subscribe to schema update when you create a subscription channel.

## Data range

Data range refers to the range of timestamps of incremental data stored in the subscription channel. The timestamp of a piece of incremental data is the time when the incremental data is applied and written to the transaction log in the database instance. By default, only data generated on the most recent day is retained in the subscription channel. DTS regularly cleans the expired incremental data and updates the data range of the subscription channel.

## Consumption checkpoint

The consumption checkpoint is the timestamp of the latest consumed incremental data that is subscribed using the downstream SDK. The SDK sends an ACK message to DTS for every piece of data that is consumed. The server updates and saves the consumption checkpoint corresponding to the SDK. When the SDK encounters an exception, the server restarts and automatically pushes the data at the latest consumption checkpoint.

# 16.Data Management (DMS)

# 16.1. Product Introduction

## 16.1.1. What is DMS?

Data Management (DMS) is a fully managed service that is provided by Apsara Stack. You can use this service to manage data, table schemas, R&D processes, R&D specifications, users, permissions, and access security.

### Supported databases

- Relational databases:
  - MySQL: ApsaraDB RDS for MySQL, PolarDB-X, MySQL databases from other cloud service providers, and self-managed MySQL databases
  - SQL Server: ApsaraDB RDS for SQL Server, SQL Server databases from other cloud service providers, and self-managed SQL Server databases
  - PostgreSQL: ApsaraDB RDS for PostgreSQL, PostgreSQL databases from other cloud service providers, and self-managed PostgreSQL databases
  - Self-managed Dameng (DM) databases
  - Self-managed Oracle databases
  - ApsaraDB for OceanBase and self-managed OceanBase databases
- NoSQL databases:
  - Redis: ApsaraDB for Redis, Redis databases from other cloud service providers, and self-managed Redis databases
  - MongoDB: ApsaraDB for MongoDB, MongoDB databases from other cloud service providers, and self-managed MongoDB databases
  - Graph Database (GDB)
- Online analytical processing (OLAP) databases:
  - AnalyticDB for MySQL
  - AnalyticDB for PostgreSQL

> **Note** Self-managed databases are databases that are installed on Apsara Stack Elastic Compute Service (ECS) instances, instances from other cloud service providers, or servers in data centers.

### Features

- DMS provides support for the entire database development process. You can design table schemas in an on-premises environment based on the design specifications. Before you publish SQL statements to an online environment, DMS can review the add, remove, modify, and query operations in the statements. Then, you can publish schemas to the specified environment as needed.

- DMS provides fine-grained access control at the database, table, or field level. You can perform all operations on databases in the DMS console. The operations can be traced and audited.
- DMS allows you to configure operation specifications and approval processes for multiple modules based on your business requirements. These modules include the schema design, data change, data export, and permission application.
- DMS integrates database development with database interaction. You can manage databases without the need to switch between database endpoints at a high frequency by using database accounts and passwords.
- DMS provides the task orchestration feature that allows you to orchestrate and schedule SQL tasks for databases. You can use this feature to perform a variety of operations with ease. For example, you can use this feature to dump historical data or generate periodical reports.

# 16.1.2. 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

## Unified operations and comprehensive audits

- After you adds 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 serval operations on databases to explore the value of data. These operations include transferring historical data and generating periodical reports.

## 16.1.3. 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.

## 16.1.4. Features

### 16.1.4.1. Data assets

DMS provides the following features that you can use to manage data assets: global asset query, sensitive data management, category, and instance management.

- Global asset query

  You can query data assets such as instances, databases, tables, and functions.

- Sensitive data management

  You can manage all fields that are classified as sensitive or confidential in a centralized manner. You can configure encryption algorithms and de-identification algorithms for sensitive and confidential fields and adjust security levels of the fields. This enhances the control of sensitive data.

- Category

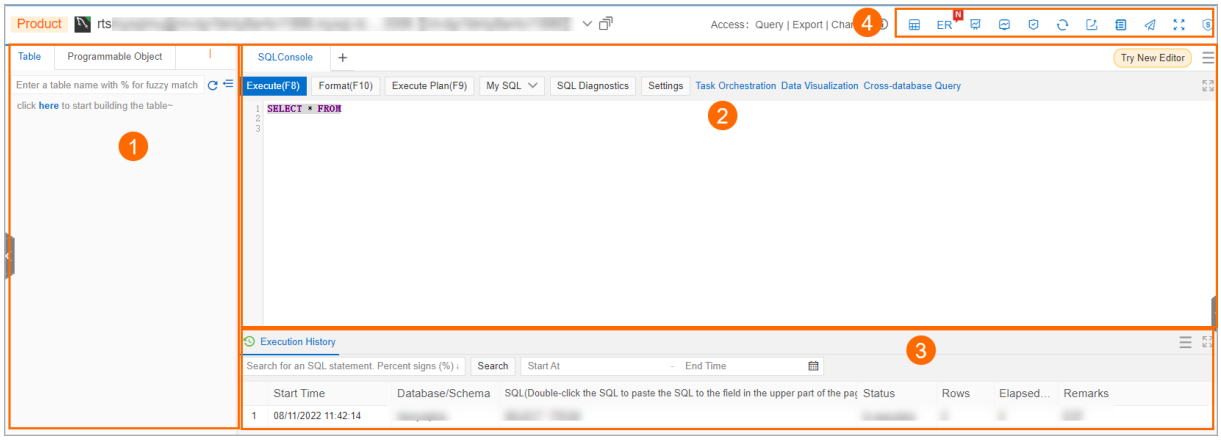
  You can add instances, databases, and tables to different categories. This helps DMS administrators, developers, and O&M engineers to better manage and use tables.

- Instance management

  You can view, enter, and edit instance information. You can grant and revoke permissions on instances. You can also disable, enable, or delete instances. The access control feature ensures that only authorized users can access instances and databases.

### 16.1.4.2. SQLConsole

You can execute various types of SQL statements in the SQLConsole of DMS. The SQLConsole allows you to add, delete, modify, and query data in the specified database in a visualized manner. You can use the SQLConsole in scenarios such as data query and data development.




| No. | Section |
| --- | --- |
| ① | Visual operation section |
| ② | SQL execution section |
| ③ | Execution result section |
| ④ | Extended feature section |

- In the visual operation section, you can view all tables, fields, and indexes of the current database. You can also right-click a table to modify the table schema, import data, or export data. You can create, view, execute, and manage programmable objects such as views, stored procedures, functions, triggers, and events.
- In the SQL execution section, you can write, format, and execute SQL statements. You can also modify or update result sets. This section provides the intelligent SQL completion prompt feature and allows you to add and manage frequently used SQL statements.
- In the execution result section, you can view the execution results and execution history. The execution results are displayed in the table form. You can export the results to Excel or text files and generate INSERT scripts for result sets. You can also copy and download a single row or multiple rows of a result set.
- Extended features
  - Table list: You can view table schemas and indexes, manage tables at the row level, and perform table-related operations. For example, you can apply for permissions on tables, apply to be data owners, adjust data owners, export table creation statements, export table schemas, and adjust the security levels of fields.
  - Metadata synchronization: You can collect the latest metadata of a database, such as tables, fields, indexes, and programmable objects. This helps you manage permissions on tables, fields, and programmable objects based on different security levels.
  - Export: You can export the data, schemas, or table creation statements of a database.
  - Schema version management: DMS automatically records table creation statements for each schema version. You can obtain the differences between two versions and generate scripts to fix the inconsistencies. This helps you effectively manage different schema versions.
  - Operation audit: You can audit operations such as SQL queries, tickets, and logon records for a single database. This helps you troubleshoot database issues.

- Risk audit: You can identify the risks about metadata, sensitive data, and SQL statements to effectively improve the security and stability of databases.
- Super SQL mode: DMS administrators or database administrators (DBAs) can enable this mode. In this mode, all SQL statements that you submit in the SQLConsole are directly executed without the limit of security rules. This mode applies to O&M and emergency handling.

# 16.1.4.3. Database development

This topic describes the database development operations that you can perform by submitting tickets.

## Schema changes

- Schema design

  You can follow the settings of an R&D process to design schemas that meet R&D specifications for the specified databases and tables. You can customize R&D processes for different lines of business based on your business requirements to ensure the consistency of schemas among multiple environments, such as the development environment, test environment, and production environment.

- Schema synchronization

  You can compare the schemas of different databases and generate scripts that can be executed in a specified database to fix inconsistencies. You can use this feature to compare and synchronize the schemas of databases in different environments, such as databases in a production environment and a test environment, databases in different test environments, or databases in different production environments.

- Shadow table synchronization

  Data Management (DMS) can automatically create a shadow table based on the schema of a source table. DMS attaches a prefix or a suffix to the name of the source table to generate the name of the shadow table. You can use this feature for end-to-end stress testing.

- Empty database initialization

  You can synchronize the schemas of the source database to an empty database that contains no table. This way, schemas can be synchronized between databases with ease. You can use this feature to synchronize the schemas of databases that are deployed across multiple regions and units.

- Table consistency repair

  You can compare schemas of a table in different environments, such as the test environment and production environment, find the differences, and execute scripts in the specified environment to fix the inconsistencies. You can also use the table consistency repair feature to ensure the consistency between logical databases. For example, you can select a table as the baseline table and check whether other table shards have the same schema as the baseline table.

## Data changes

- Regular data changes

You can execute SQL statements such as INSERT, UPDATE, DELETE, and TRUNCATE statements to update data in databases. You can perform regular data changes in scenarios such as data initialization, historical data deletion, troubleshooting, and feature testing.

- Lockless changes
  - Lockless change based on DML: This feature allows you to split a single SQL statement into multiple batches for execution based on the powerful DMS engine. This can better meet the needs of large-volume data changes, such as historical data cleanup and field updates in a full table. This also ensures execution efficiency and reduces the impact on database performance and database space.
  - Lockless change based on DDL: This feature allows you to change schemas without the need to lock tables. You can use this feature to prevent business interruption that can be caused by table locking during schema changes. You can also use this feature to prevent latency in synchronization between primary and secondary databases that occurs when schemas are changed by using native online DDL operations.
- Historical data cleanup

  You can periodically delete historical data based on lockless changes to prevent historical data accumulation from affecting the stability of the production environment.
- Programmable objects

  You can change stored procedures. DMS provides a standardized management process to control the change operations.
- Data import

  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.

## Data export

- SQL result set export

  You can write and execute SQL query statements and export the result sets.
- Database export

  You can export a database or specific tables. You can also export only the schemas of a database or specific tables or export table data to extract data for analysis.

## SQL review

DMS reviews submitted SQL statements based on security rules and provides optimization suggestions. This feature eliminates SQL statements that do not use indexes or do not conform to database development standards. This also helps protect against SQL injection attacks. You can customize SQL specifications in security rules. For example, you can specify that a table must have a primary key, the data type of a primary key column must be restricted, or the number of primary key columns must be restricted.

## Environment construction

- Database cloning

You can replicate data by cloning databases. You can use this feature to synchronize full databases or initialize databases across different environments, such as the development environment and test environment.

- Test data generation

  DMS provides powerful algorithm engines that you can use to generate a large amount of test data, such as random values, region names, and virtual IP addresses. This greatly improves the efficiency of preparing test data.

### R&D space

- DevOps: You can use the DevOps feature to customize R&D processes and manage the quality of these processes. This effectively ensures the implementation of R&D processes, reduces accidental operations, and protects data security. The DevOps feature allows you to advance an R&D process stage by stage. You can create specific types of tickets in each iterative stage. This facilitates collaborative development and improves R&D efficiency.
- My tickets: You can use this feature to view and approve pending tickets, view the tickets that you have submitted and processed, and view ticket status and ticket details.

## 16.1.4.4. DTS

This topic describes the features that the Data Transmission Service (DTS) module of Data Management (DMS) provides for data migration, change tracking, data integration, data development, and data application.

- Data migration

  You can use DTS to migrate data between homogeneous and heterogeneous data sources. This feature is suitable for the following scenarios: data migration to Alibaba Cloud, data migration between instances in Alibaba Cloud, and database splitting and scale-out.

- Change tracking

  You can use DTS to track incremental data from ApsaraDB RDS for MySQL instances, PolarDB for MySQL instances, PolarDB-X instances, self-managed MySQL databases, and self-managed Oracle databases in real time. Then, you can consume the tracked data as needed. This feature is suitable for the following scenarios: cache updates, business decoupling and asynchronous data processing, real-time data synchronization between heterogeneous databases, and real-time data synchronization that involves extract, transform, load (ETL) operations.

- Data integration
  - Data synchronization

    You can use DTS to synchronize data between data sources in real time. This feature is suitable for the following scenarios: active geo-redundancy, geo-disaster recovery, zone-disaster recovery, cross-border data synchronization, cloud-based business intelligence (BI) systems, and real-time data warehousing.

  - Batch processing

    This feature provides a low-code development tool that you can use to develop data processing tasks. You can combine various task nodes to form a data flow and configure periodic scheduling to process or synchronize data. You can use this feature to process complex big data in scenarios such as refined enterprise operations, digital marketing, and intelligent recommendation.

  - Streaming ETL

    You can extract, transform, process, and load streaming data. This feature helps enterprises navigate real-time data processing and computing scenarios and promote the digital transformation of enterprises.

- Task orchestration

  You can use the task orchestration feature to orchestrate various types of tasks and then schedule and run the tasks. You can create a task flow that consists of one or more task nodes. This helps schedule tasks in complex scenarios and improve the efficiency of data development. DMS supports the following types of task nodes: Cross-database SQL, Single-instance SQL, Table status check, and Data Lake Analytics (DLA) serverless Spark.

- Data application
  - Data service

    You can use the data service feature to export data that is managed by DMS to external environments. The data service feature allows you to export the data of a specific row or column to external environments. To export the data of the specific row, specify filter conditions in SQL statements. To export the data of the specific column, specify the fields that allow queries. Compared with data export of a whole table, this minimizes data exposure and ensures data security.

  - Data visualization

    The data visualization feature provides a three-layer model that you can use to visualize data in various forms. The three layers are datasets, charts, and dashboards or big screens. You can display the results of a single SQL statement in regular charts such as line charts, pie charts, column charts, circular charts, table charts, dual Y-axis charts, and funnel charts. Then, you can combine these charts in the specified layout on a dashboard or big screen to visualize business data based on your analysis idea or methodology. Data visualization helps you gain insights into your business and make informed business decisions.

- Advanced Database & Application Migration (ADAM)
  - Database evaluation

    The database evaluation feature performs intelligent evaluation and analysis based on the data that is collected from the source database. Then, the feature generates a report that contains multiple factors, such as the database migration solution, compatibility of the destination database, migration risks, application transformation suggestions, and migration costs.

- Database transformation and migration

  This feature provides intelligent tools for database transformation and migration. This facilitates source database comparison, and schema migration and modification based on analysis results.

- Application evaluation and transformation

  This feature provides application transformation items to migrate databases, such as focus areas and SQL statements, analyzes application frameworks, and provides architecture migration guidelines. This way, you can transform your application in an efficient manner.

- Migration lab
  - The periodic SQL collection feature periodically collects SQL statements from a database and automatically combines the SQL statements.
  - The SQL adapter feature can convert non-compatible SQL statements that are migrated from Oracle to PolarDB for Oracle in real time. This feature asynchronously records all SQL statements that need to be converted.
  - The SQL comparison and test platform can compare the execution results of an SQL statement in the source and destination databases. The results show the differences in execution time, number of returned rows, and result set verification.

- SQL conversion

  This feature converts the SQL statements of the source database into the SQL statements that can be used in the destination database. This reduces the code workload of developers during database migration.

# 16.1.4.5. Security and specifications

This topic describes the features that the security and specifications module of Data Management (DMS) provides to ensure database security and compliance.

- Permissions

  A tenant can apply for specific operation permissions on a specific database instance, database, table, column, or row. After the application is approved, the tenant can perform operations within the permissions. The operation permissions include the query, change, and export permissions.

  - Query permissions: the permissions to execute SQL statements on the SQL Console tab.
  - Change permissions: the permissions to submit tickets to change data or synchronize data in a database or table. You cannot change data without approval.
  - Export permissions: the permissions to submit tickets to export data. You cannot export data without approval.

  The control permissions include instance logon permissions, performance view permissions, database permissions, table permissions, sensitive column permissions, data row permissions, and programmable object permissions.

- Security rules

Security rules use a domain-specific language (DSL) to implement fine-grained control over databases. When you query, export, or change data in the DMS console, security rules are used to regulate your behavior. This helps create database operation specifications and R&D processes.

  - Security rules integrate R&D processes, R&D specifications, and approval processes. You can use security rules to coordinate DMS features and allow multiple online developers to collaboratively manage databases.
  - Security rules support a variety of SQL engines. You can customize security rules to check and manage SQL statements.
  - Security rules provide a powerful approval feature. You can customize approval processes based on different user behavior.

- Approval processes

  Approval processes are associated with security rules. You can select or configure different approval processes for different user behaviors. A single approval node constitutes an approval stage. DMS allows you to add, modify, and delete approval nodes. You can add one or more approvers to an approval node.

- IP address whitelists

  IP address whitelists allow you to effectively control the range of users who can access DMS. You can use IP address whitelists to allow only the users from specific trusted network environments to access DMS.

- Operation audit

  You can query tickets, logon records, operation logs, and the SQL statements that were executed on the SQL Console tab. This helps you troubleshoot database issues and collect data for operation audit.

- Sensitive data management

  You can use this feature to effectively detect and protect sensitive data assets in your enterprise. This prevents sensitive data from being abused or leaked.

  - Sensitive data list

    You can adjust the sensitivity level and data masking rules of fields, grant permissions on specific fields to a user, and revoke permissions on the fields from a user.

  - Sensitive data detection

    You can view the sensitive data detection tasks and rules of a database instance. DMS has built-in sensitive data detection rules that you can use to ensure compliance with various laws and regulations. If the built-in sensitive data detection rules cannot meet your business requirements, you can create custom sensitive data detection rules.

  - Data masking management

    This feature provides a built-in data masking rule that masks the entire value of a field. You can create custom data masking rules based on the built-in data masking algorithms.

# 16.1.4.6. Solution

This topic describes the T+1 full data snapshot feature provided by the Solution module of Data Management (DMS).

The T+1 full data snapshot feature allows you to create snapshots for specified tables every hour or day on a T+1 basis. This way, you can view the statistics on data by hour, day, or month.

### Scenarios

The T+1 full data snapshot feature is commonly used to store your business data to data warehouses. This feature allows you to synchronize full data on an hourly or daily basis. This way, you can view the statistics on data by hour, day, or month. You can use this feature in the following scenarios:

- Record the daily account balance for bill queries and account reconciliation in an accounting system.
- Record the daily price of a product to check whether the product is sold at the lowest ever price or whether the product is suitable for promotion.
- Collect statistics and calculate the total amount of orders on the previous day to obtain up-to-date information about business operations.

## 16.1.4.7. O&M

This topic describes the O&M features of DMS. DMS provides the following O&M features: user management, task management, configuration management, and database grouping.

- User management

  You can add and delete DMS users, control user permissions, and grant permissions on instances, databases, tables, rows, and sensitive columns.

- Task management

  You can create SQL tasks and manage existing tasks. In DMS, tasks are generated for SQL statements that are executed for various features such as normal data change, schema design, and database or table synchronization. The SQL statements that are executed in the SQLConsole are not included.

- Configuration management

  DMS administrators and DBAs can modify system configurations based on their business requirements. For example, they can enable automatic registration of cloud instance resources, enable IP address whitelists for access control, and specify whether to allow DMS administrators and DBAs to act beyond their authority to approve or reject tickets.

- Database grouping

  You can add databases that are of the same engine type and deployed in the same environment to a group. After you create a database group, you can load all databases in the group during data changes or schema design. After the ticket is submitted, the changes can take effect in all databases in the group.

## 16.1.5. 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 |
| --- | --- | --- |
| Instance owner | • Each instance has only one owner.<br>• 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.<br>• 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.<br>• 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. |

| Role | Description | Permission |
| --- | --- | --- |
| 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.<br>• 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.<br>• 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.<br>• 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.<br>• 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.<br>• 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.<br>• 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. |
| DBA | • Each instance has only one DBA.<br>• 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.<br>• 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.<br>• 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.<br>• 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.<br>• 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.<br>• Database instances are used by a small-sized group that includes multiple users. | A database account and the related password. |
| 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.<br>• This control mode allows you to produce enterprise-specific database DevOps solutions through custom design specifications and approval processes.<br>• All features that are included in the flexible management and stable change control modes are available. | • Ensure the data security of database instances.<br>• Implement strict access control over development or change workflows.<br>• 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.

# 17.Server Load Balancer (SLB)

# 17.1. Product Introduction

## 17.1.1. 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.

# 17.1.2. 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.

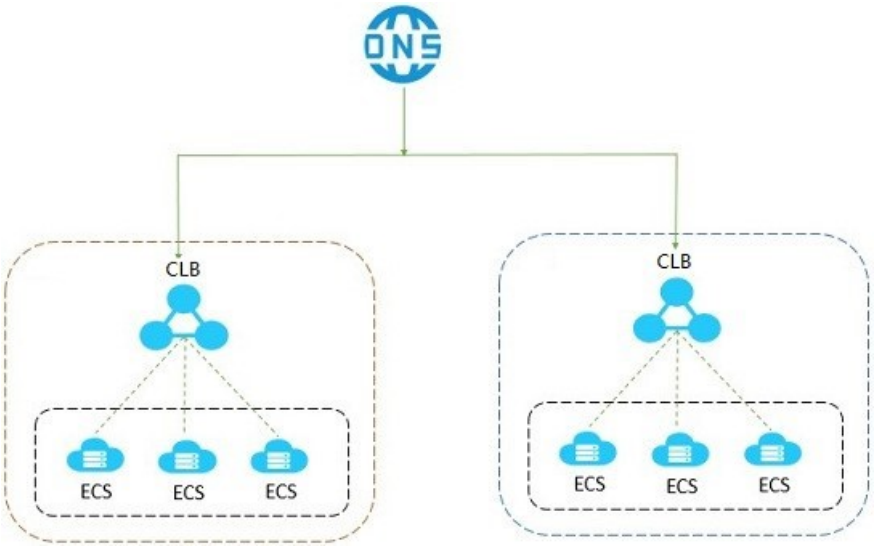
   However, we recommend that you do not deploy all ECS instances in the primary zone. You must deploy a small number of ECS instances in the secondary zone. This way, requests can be redirected to backend servers in the secondary zone when the primary zone becomes unavailable.

## 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*.

# 17.1.3. Architecture

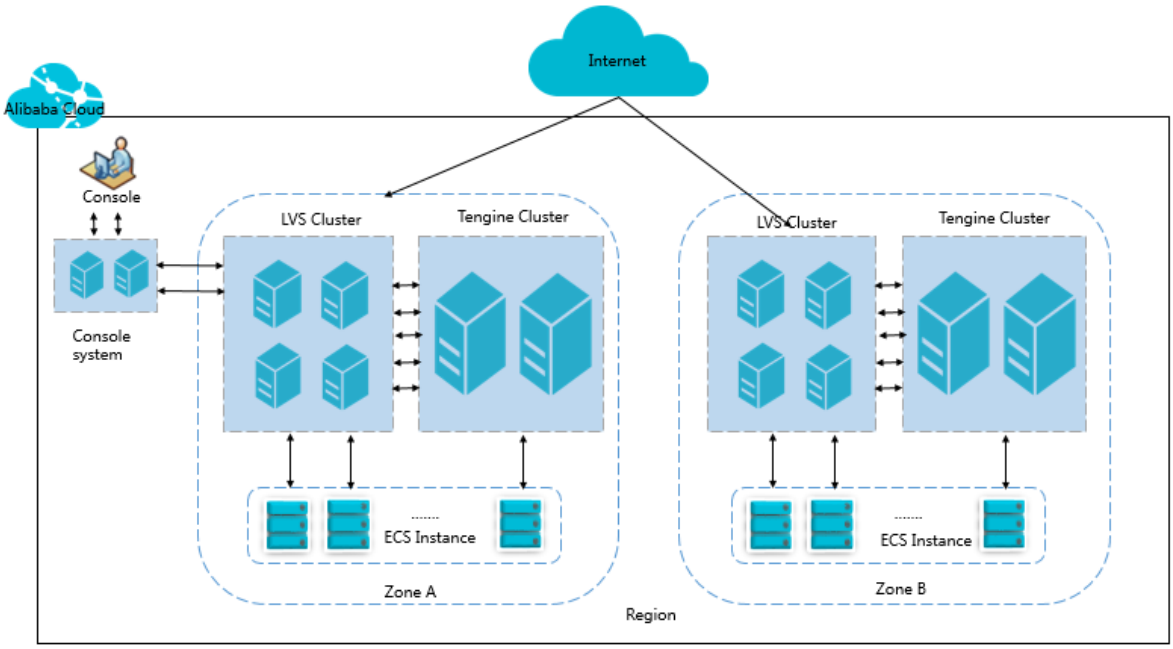
This topic describes the architecture of Server Load Balancer (SLB). SLB instances are deployed in clusters to synchronize sessions and protect backend servers from single points of failure (SPOFs). This improves redundancy and ensures service stability. SLB supports load balancing at Layer 4 for TCP 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.

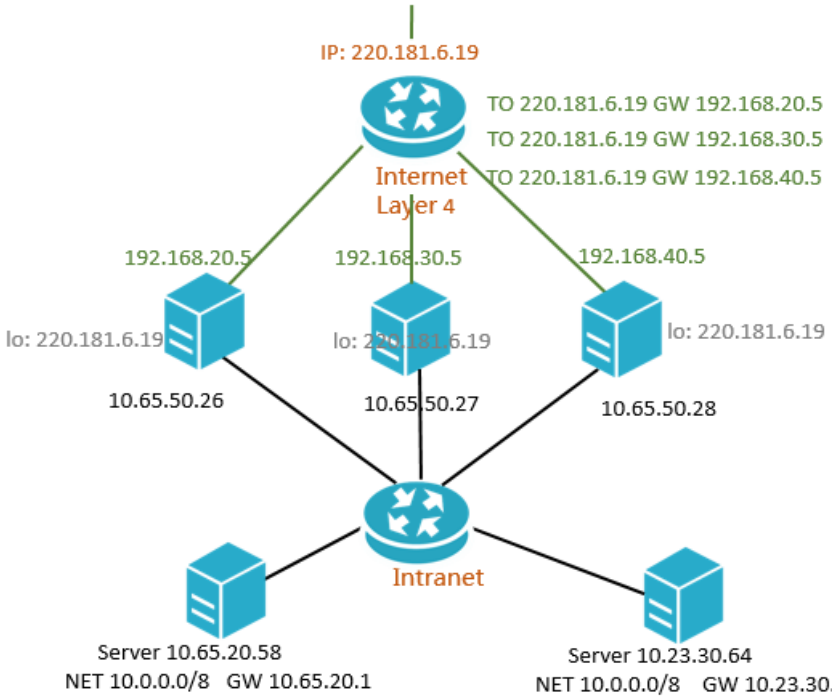
## SLB design

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.

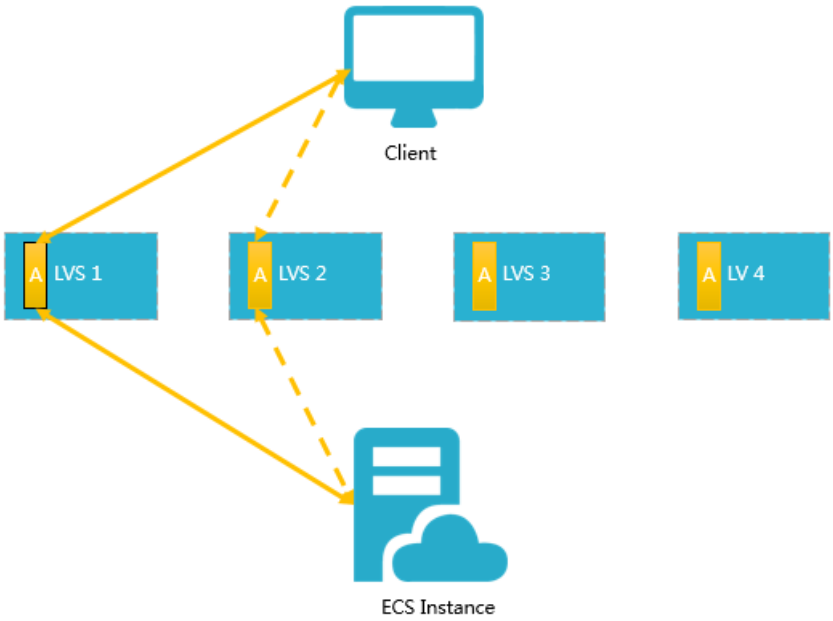


In each region, Layer 4 SLB runs in an LVS cluster that consists of multiple LVS machines, as shown in the following figure. This cluster deployment mode enhances the availability, stability, and scalability of load balancing services in abnormal cases.




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.




# 17.1.4. 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<br>SLB supports the round-robin (RR), weighted round-robin (WRR), and consistent hashing scheduling algorithms. | Y | Y<br>Note Layer 7 SLB does not support the consistent hashing scheduling algorithm. |
| Health checks<br>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<br>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<br>SLB uses whitelists to control access to your applications. | Y | Y |
| High availability<br>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<br>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<br>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<br>Internet-facing SLB instances can forward requests from IPv6 clients. | Y | Y |
| Certificate management<br>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<br>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<br>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 |

# 17.1.5. 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.

## 17.1.6. 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 |

## 17.1.7. 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. |

| Term | Description |
| --- | --- |
| 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.<br>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. |
| vServer group | A group of ECS instances that process distributed requests.<br>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. |

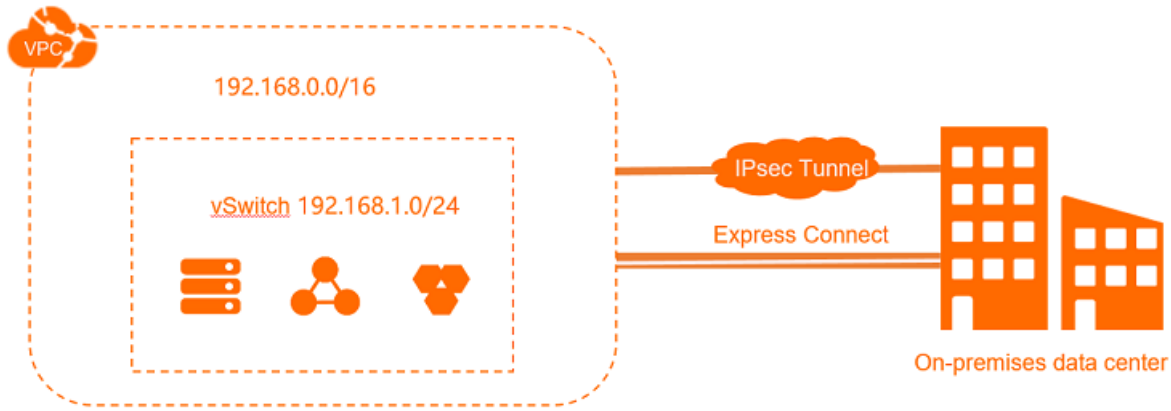
# 18.Virtual Private Cloud (VPC)

## 18.1. Product Introduction

### 18.1.1. What is a VPC?

A virtual private cloud (VPC) is a private network in the cloud. You can configure the CIDR block, route tables, and gateways of your VPC. You can use Alibaba Cloud services in a VPC, such as Elastic Compute Service (ECS), Server Load Balancer (SLB), and ApsaraDB RDS.

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 VPC consists of one vRouter, at least one private CIDR block, and at least one vSwitch.




- 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 the *Plan and design a VPC* topic in *User Guide*.

| CIDR block | Number of available private IP addresses (excluding system reserved IP addresses) |
| --- | --- |
| 192.168.0.0/16 | 65,532 |
| 172.16.0.0/16 | 65,532 |

- vRouters

  A vRouter is the hub of a VPC. As a core component, it connects the vSwitches in a VPC and serves as a gateway between a VPC and other networks. After you create a VPC, the system automatically creates a vRouter. Each vRouter is associated with a route table.

  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 vSwitches to create one or more subnets for the VPC. vSwitches in the same VPC can communicate with each other. You can deploy your applications in vSwitches that belong to different zones to improve service availability.

  For more information, see the *Create a vSwitch* topic in *User Guide*.

# 18.1.2. Benefits

This topic describes the benefits of virtual private clouds (VPCs). VPCs are secure, reliable, flexible, easy to use, and scalable.

## Security and reliability

Each VPC is identified by a unique tunnel ID, which corresponds to a virtual network. Different VPCs are isolated by tunnel IDs:

- Similar to a traditional network, you can create vSwitches and vRouters to divide a VPC into multiple subnets. Elastic Compute Service (ECS) instances in the same subnet use the same vSwitch to communicate with each other, while ECS instances in different subnets use vRouters to communicate with each other.
- VPCs are completely isolated from each other. Cloud resources in different VPCs can communicate with each other by using elastic IP addresses (EIPs) or NAT IP addresses.
- The IP packets of an ECS instance are encapsulated by using the tunneling technology. Therefore, information at the data link layer (the MAC address) of the ECS instance is not transferred to the physical network. This way, ECS instances in different VPCs are isolated at Layer 2.
- ECS instances in a VPC use security groups and firewalls to control inbound and outbound traffic at Layer 3.

## Flexible management

You can use security group rules and access control lists (ACLs) to manage inbound and outbound traffic to cloud resources in a VPC in a flexible manner.

## Ease of use

You can easily create and manage VPCs in the VPC console. When you create a VPC, the system automatically creates a vRouter and a route table for the VPC.

### High scalability

You can create different subnets in a VPC to deploy different services. Additionally, you can connect a VPC to a data center or another VPC to extend the network architecture.

## 18.1.3. Basic architecture

Virtual private clouds (VPCs) are isolated from each other based on a tunneling technology. Each VPC is identified by a unique tunnel ID, which corresponds to a virtualized network.

### Background information

The development of cloud computing technologies leads to higher requirements for virtual networks, such as scalability, security, reliability, privacy, and robust connectivity performance. This speeds up the development of various technologies about network virtualization.

In earlier solutions, virtual and physical networks are merged to generate a flat network architecture, such as large-scale Layer 2 networks. As the scale of virtual networks grows, these solutions encounter problems such as Address Resolution Protocol (ARP) spoofing, broadcast storms, and host scanning. To resolve these problems, various network isolation technologies emerged. With these technologies, physical networks are isolated from virtual networks. One of these technologies adopts virtual local area networks (VLANs) to isolate networks. However, VLANs support at most 4096 VLAN IDs and do not apply to large-scale networks.

### How VPC works

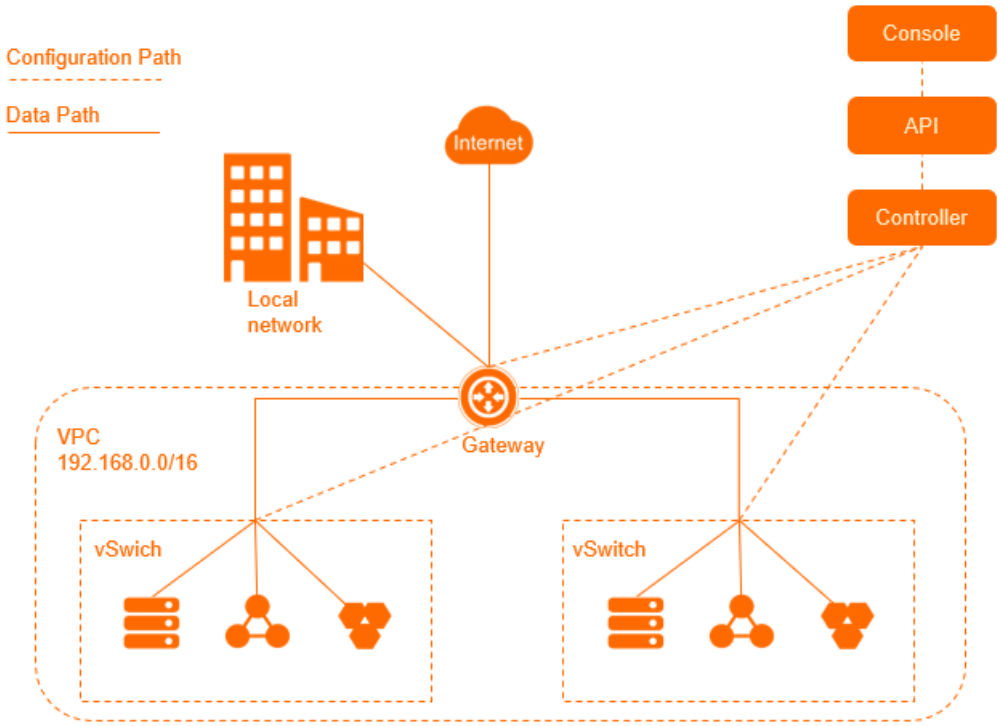
VPCs are isolated from each other based on a tunneling technology. Each VPC is identified by a unique tunnel ID, which corresponds to a virtual network.

- Data packets are encapsulated with a unique tunnel ID and transmitted over a physical network between Elastic Compute Service (ECS) instances in a VPC.
- Data packets transmitted over ECS instances in different VPCs have different tunnel IDs. Therefore, ECS instances in different VPCs cannot communicate with each other.

Alibaba Cloud developed VPCs that are integrated with gateways and vSwitches by adopting the tunneling and Software Defined Network (SDN) technologies.

### Logical architecture of VPCs

A VPC contains a gateway, a controller, and one or more vSwitches, as shown in the following figure. The vSwitches and gateway form a data path where data is transferred. The controller uses a protocol developed by Alibaba Cloud to form a configuration path. The data path is isolated from the configuration path. vSwitches in VPCs are distributed nodes while gateways and controllers are deployed in clusters in multiple data centers. All VPC connections support disaster recovery, which ensures the high availability.




# 18.1.4. Features

A virtual private cloud (VPC) is a private network in the cloud. VPCs are logically isolated from each other. This topic describes the features of VPCs.

## Custom private networks

You can specify custom private networks for your VPCs. When you create a VPC and a vSwitch, you can specify private CIDR blocks for them. In addition, you can create multiple subnets for a VPC and deploy services in different subnets to improve service availability.

## Custom routes

You can add custom routes to a 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 multiple route entries match the destination IP address, the route entry with the longest subnet mask prevails and is used to determine the next hop. This ensures that the traffic is routed to the most precise destination.

## Various connection methods

A VPC provides various connection methods. You can connect a VPC to the Internet, a data center, or another VPC.

- Connect a VPC to the Internet

  You can connect a VPC to the Internet by associating elastic IP addresses (EIPs) with resources in the VPC or configuring NAT gateways. This way, resources in the VPC can communicate with the Internet.

- Connect a VPC to another VPC

  You can establish high-performance and secure connections between VPCs by creating a pair of router interfaces.

- Connect a VPC to a data center

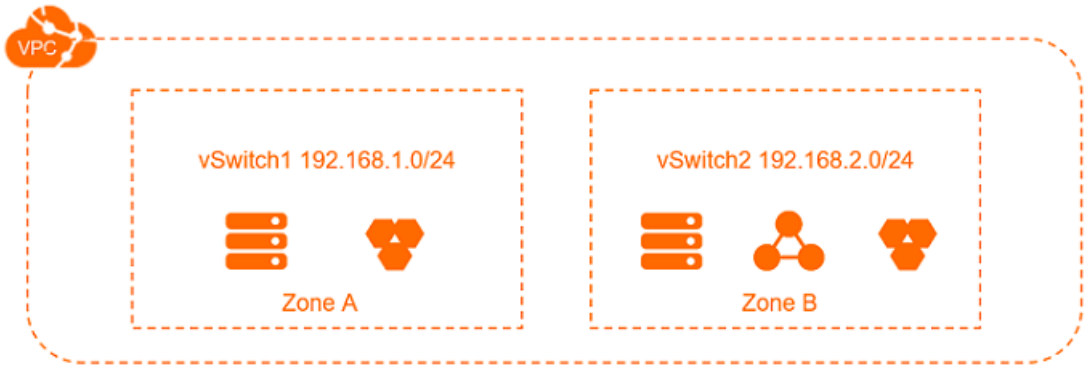
  You can connect a VPC to a data center through Express Connect circuits. This allows you to smoothly migrate on-premises application to the cloud.

# 18.1.5. Use scenarios

Virtual private clouds (VPCs) are virtual networks that are isolated from each other. VPCs support flexible configurations to meet the requirements of different scenarios.

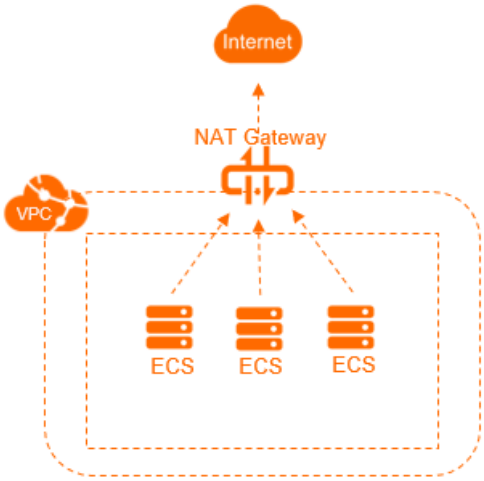
## Deploy applications in a safe manner

You can deploy applications in a VPC to provide services to external networks. To control access to the applications over the Internet, you can create security group rules and configure whitelists. You can also isolate application servers from databases to implement access control. For example, you can deploy web servers in a subnet that can access the Internet, and deploy databases in another subnet that cannot access the Internet.



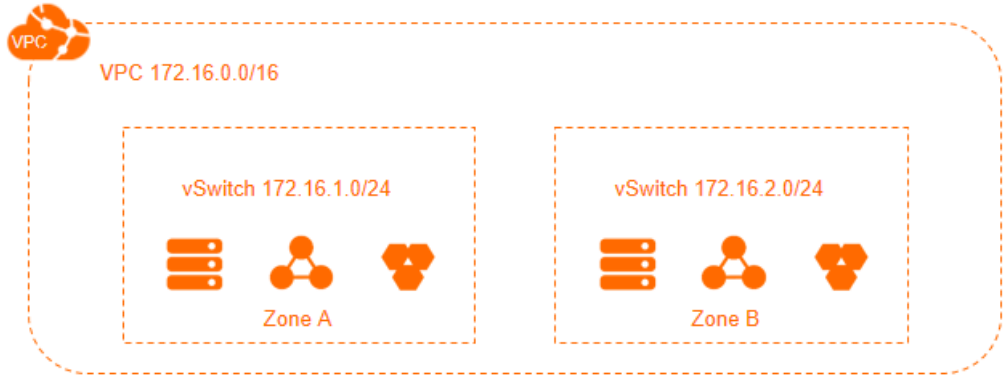

## Deploy applications that require access to the Internet

You can deploy applications that require access to the Internet in a subnet of a VPC and use an Internet NAT gateway to route network traffic. You can configure SNAT entries 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 elastic IP addresses (EIPs) specified in the SNAT entries to prevent attacks from the Internet.



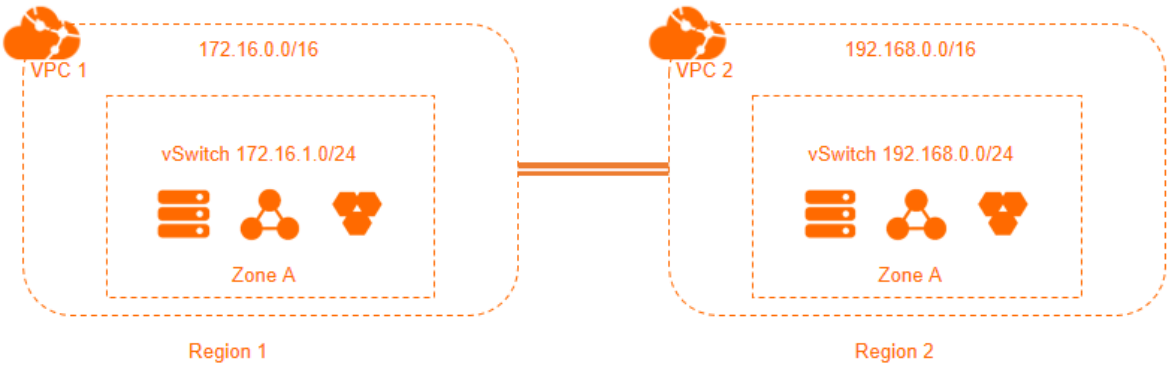

## Implement cross-zone disaster recovery

You can create one or more vSwitches to create one or more subnets for the VPC. 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.



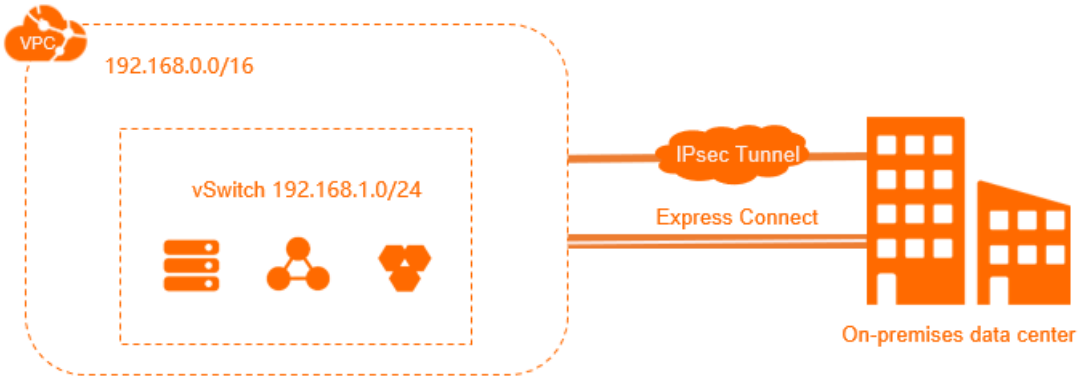

## Isolate business systems

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 establish a dedicated connection between a VPC and your data center. This allows you to seamlessly migrate the applications in your data center to the cloud. You do not need to change the access method for the applications.




# 18.1.6. Background information

This topic describes basic terms about Virtual Private Cloud (VPC).

| Term | Description |
| --- | --- |
| VPC | A VPC is a private network 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 consists of route entries in a vRouter. |
| Route entry | Each item in a route table is a route. 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. |

# 18.1.7. 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 | Limit |
| --- | --- |
| Maximum number of VPCs that can be created in each region | 200 |
| Maximum 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. |
| Maximum number of secondary IPv4 CIDR blocks that can be created in each VPC | 1 |
| Maximum number of private IP addresses that can be used by cloud resources in each VPC | • IPv4 addresses: 60,000<br>• IPv6 addresses: 2,000<br>Note<br>• If an Elastic Compute Service (ECS) instance has only one private IP address, the ECS instance uses only one network address.<br>• If an ECS instance is associated with multiple elastic network interfaces (ENIs), or multiple IP addresses are assigned to an ENI, the Maximum number of network addresses used by the ECS instance equals the total Maximum number of the IP addresses assigned to the ENIs that are associated with the ECS instance. |

## Limits on vRouters and route tables

| Item | Limit |
| --- | --- |
| Maximum number of vRouters that can be created in each VPC | 1 |
| Maximum number of custom route entries that can be created in each route table | 48 |

| Item | Limit |
| --- | --- |
| Maximum number of tags that can be added to each route table | 20 |

## Limits on network access control lists (ACLs)

| Item | Limit |
| --- | --- |
| Maximum number of network ACLs that can be created in each VPC | 200 |
| Maximum number of network ACLs that can be associated with a vSwitch | 1 |
| Maximum number of rules that can be added to a network ACL | • Inbound rules: 20<br>• Outbound rules: 20 |

## Limits on high-availability virtual IP addresses (HAVIPs)

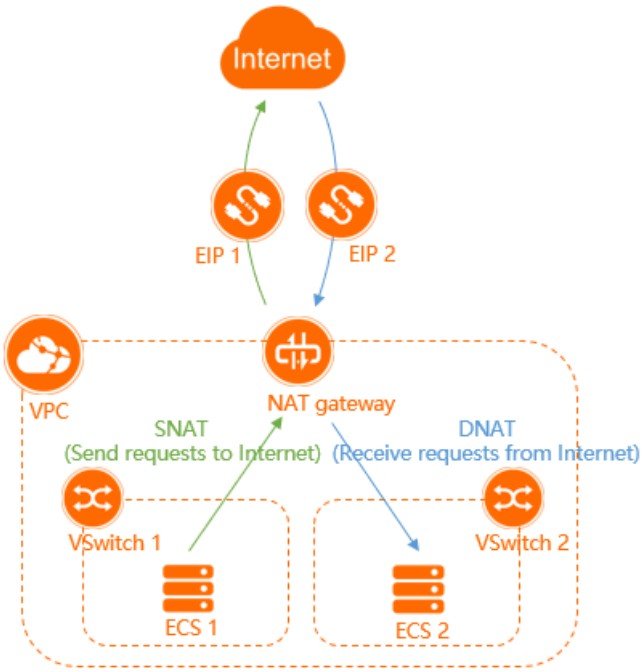
| Item | Limit |
| --- | --- |
| Maximum number of HAVIPs that can be created in each VPC | 5 |
| Maximum number of HAVIPs that can be associated with each ECS instance | 5 |
| Maximum number of HAVIPs that can be associated with each ENI | 5 |
| Maximum number of ECS instances that can be associated with each HAVIP | 2 |
| Maximum number of ENIs that can be associated with each HAVIP | 2 |
| Maximum number of route entries that point to an HAVIP in each VPC | 5 |
| Whether HAVIPs support broadcasting or multicasting | Not supported<br>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. |

# 19.NAT Gateway

## 19.1. Product Introduction

### 19.1.1. What is NAT Gateway?

NAT gateways are enterprise-class gateways that provide the SNAT and DNAT features. Each NAT gateway provides a throughput capacity of up to 10 Gbit/s. NAT gateways also support cross-zone disaster recovery.




#### Overview

A NAT gateway works as expected only after an elastic IP address (EIP) is associated with the NAT gateway. After you create a NAT gateway, you can associate an EIP with the NAT gateway.

NAT gateways provide the SNAT and DNAT features. The following table describes the features.

| Feature | Description |
| --- | --- |
| SNAT | SNAT allows Elastic Compute Service (ECS) instances that are deployed in a virtual private cloud (VPC) to access the Internet when no public IP addresses are assigned to the ECS instances. |
| DNAT | DNAT maps the EIPs that are associated with a NAT gateway to ECS instances. This way, the ECS instances can provide Internet-facing services. |

### 19.1.2. Description

NAT Gateway provides the SNAT and DNAT features.

### SNAT

NAT gateways support the SNAT feature. SNAT enables Elastic Compute Service (ECS) instances that do not have public IP addresses assigned in a virtual private cloud (VPC) to access the Internet.

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 initiate requests. However, ECS instances in the VPC cannot be accessed over the Internet. SNAT shields the ports that the ECS instances use to communicate with the Internet. This protects the ECS instances from external attacks.

### DNAT

NAT Gateway provides the DNAT feature to map public IP addresses to ECS instances. This way, ECS instances can provide services over the Internet.

## 19.1.3. Benefits

NAT Gateway features easy configuration, high performance, high availability, and on-demand purchase.

### Easy configuration

A NAT gateway is an enterprise-grade Internet gateway that provides the SNAT and DNAT features. NAT gateways are reliable, flexible, and easy-to-use. NAT gateways save you the trouble of building an Internet gateway by yourself.

### High performance

Alibaba Cloud NAT gateways are distributed gateways that use the software-defined networking (SDN) technology. Each NAT gateway provides a throughput capacity of up to 10 Gbit/s, and can serve a large number of Internet applications.

### High availability

You can deploy a NAT gateway across zones to implement high availability. Failures in one zone do not affect service continuity.

### On-demand purchase

You can modify the specifications of NAT gateways, adjust the number of EIPs, and modify bandwidth of EIPs based on your business requirements.

## 19.1.4. Use scenarios

NAT gateways allow Elastic Compute Service (ECS) instances in virtual private clouds (VPCs) to access the Internet and receive requests from the Internet.

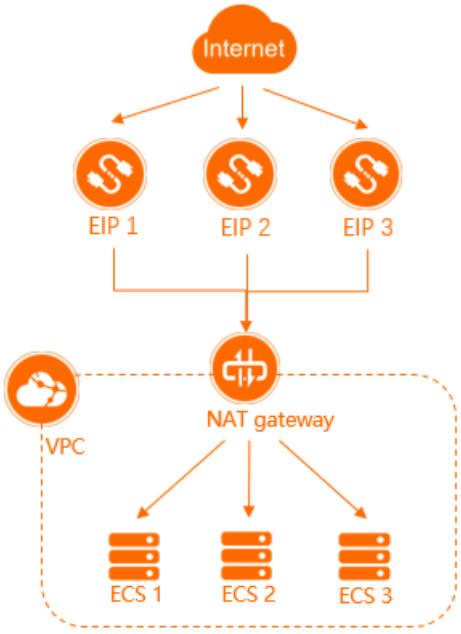
### Configure SNAT to enable ECS instances to access the Internet

You can create a NAT gateway for a VPC, associate an elastic IP address (EIP) with the NAT gateway, and then create an 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.

### Configure DNAT to provide Internet-facing services

You can create a NAT gateway for a VPC, associate EIPs with the NAT gateway, and then create a DNAT entry on the NAT gateway. This way, ECS instances in a VPC can receive requests from the Internet through port mapping or IP mapping.

> **Note** Descriptions of port mapping and IP mapping:
>
> - Port mapping: A NAT gateway forwards requests destined for an EIP to the specified ECS instance based on the specified ports and protocol.
> - IP mapping: A NAT gateway forwards all requests destined for an EIP to the specified ECS instance.




## 19.1.5. Terms

Before you use NAT gateways, make sure that you understand the terms that are described in this topic.

| Term | Description |
| --- | --- |
| NAT Gateway | NAT gateways are enterprise-class gateways that provide the SNAT and DNAT features. NAT gateways provide a maximum forwarding capacity of 10 Gbit/s, ensure high availability across regions, and support cross-zone disaster recovery. |

| Term | Description |
| --- | --- |
| DNAT table | A DNAT table is used to configure the DNAT feature. The DNAT feature allows you to map a public IP address of a NAT gateway to an Elastic Compute Service (ECS) instance through port mapping or IP mapping. |
| SNAT table | An SNAT table is used to configure the SNAT feature. You can add an SNAT entry for a vSwitch or for a specified ECS instance.<br>• Add an SNAT entry for a vSwitch: All ECS instances attached to the vSwitch use the specified public IP address to access the Internet.<br>• Add an SNAT entry for an ECS instance: The ECS instance uses the specified public IP address to access the Internet. |
| Elastic IP Address (EIP) | An EIP is a public IP address that you can purchase and use as an independent resource. You can associate EIPs with the following resources that are deployed in virtual private clouds (VPCs): ECS instances, internal-facing Server Load Balancer (SLB) instances, and secondary elastic network interfaces (ENIs). You can also associate EIPs with NAT gateways and high-availability virtual IP addresses (HAVIPs). A NAT gateway must be associated with an EIP to provide services. |

# 19.1.6. Limits

This topic describes the limits and quotas of NAT Gateway.

| Item | Limit |
| --- | --- |
| Maximum number of NAT gateways that can be created in a virtual private cloud (VPC) | 1 |
| Specify an elastic IP address (EIP) in both SNAT and DNAT tables | Not supported |
| Maximum number of DNAT entries that can be added to a NAT gateway | 100 |
| Maximum number of SNAT entries that can be added to a NAT gateway | 40 |
| Maximum number of EIPs that can be specified in an 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<br>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. |

| Item | Limit |
| --- | --- |
| Whether the bandwidth of a vSwitch is limited by the maximum bandwidth of the EIPs in the SNAT entry that is created for the vSwitch | No |
| Maximum number of EIPs that can be associated with a NAT gateway | 20 |
| Maximum number of concurrent connections supported by a NAT gateway when ECS instances without public IP addresses in a VPC access the same destination IP address and port through the NAT gateway | N × 55,000. N indicates the number of EIPs that are specified in the SNAT entry. |

# 20.Elastic IP Address

# 20.1. Product Introduction

## 20.1.1. EIP overview

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 associated ECS instance | No | No |

### 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 an 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.

# 20.1.2. Limits

Before you use elastic IP addresses (EIPs), make sure that you understand the limits.

## General limits

- You can associate an EIP only with 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 reasons, 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 Elastic Compute Service (ECS) instance

- 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 or an EIP.
- Each ECS instance can be associated only with one EIP. If you want an ECS instance to use multiple EIPs, you can attach secondary elastic network interfaces (ENIs) that are associated with EIPs to the ECS instance.

  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

The NAT gateway and the EIP must belong to the same region.

## Limits on associating an EIP with a Server Load Balancer (SLB) instance

- 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 a secondary ENI

- 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 is determined by the number of private IP addresses of the secondary ENI.

# 21.Express Connect

# 21.1. Product Introduction

## 21.1.1. What is Express Connect?

Apsara Stack Express Connect allows you to establish private, flexible, stable, and secure connections between virtual private clouds (VPCs). Network traffic does not traverse the Internet, which prevents data breaches and ensures network stability.

Express Connect allows you to connect VPCs within the same region and account. You can also connect VPCs across different regions and accounts.

### Benefits

Express Connect provides the following benefits:

- High-speed connections

  Powered by the network virtualization technology of Apsara Stack, Express Connect allows networks to communicate with each other through direct, private, and high-speed connections. Network traffic does not traverse the Internet. The impact of distance on network performance is minimized to ensure low-latency and high-bandwidth communication.

- Stability and reliability

  Express Connect provides services based on the high-quality infrastructure of Apsara Stack. This guarantees stable and reliable communication between networks.

- Security

  Express Connect allows you to virtualize your networks and enables communication based on the infrastructure provided by Apsara Stack. Networks owned by different accounts are independent of each other. Data packets are exchanged through private connections to prevent beaches.

- On-demand purchase

  The maximum bandwidth of Express Connect circuits varies. You can set the maximum bandwidth based on your business requirements.

### Differences between Express Connect and Internet connections

A VPC is a logically isolated network. Network traffic between VPCs or between VPCs and data centers are transferred through separate connections.

Compared with Internet connections, Express Connect provides higher performance and security. The following table describes the differences.

| Item | Connect VPCs through Internet connections | Connect VPCs through Express Connect |
| --- | --- | --- |
| Network performance and availability | When network traffic is transferred over a long distance, a low network latency and a low packet loss rate cannot be guaranteed. | Connections are built based on the infrastructure provided by Apsara Stack, which offers higher quality and availability. |
| Costs | You are charged a high Internet bandwidth fee or data transfer fee. | The cost for cross-region communication is minimized.<br>You are not charged for connecting VPCs that belong to the same region. |
| Security | Network traffic is transferred over the Internet. Therefore, data breaches may occur. | Networks are virtualized on Apsara Stack and isolated from each other to ensure high security. |

# 21.1.2. Benefits

**A virtual private cloud (VPC) is a logically isolated network. Network traffic between VPCs is transferred through separate connections.**

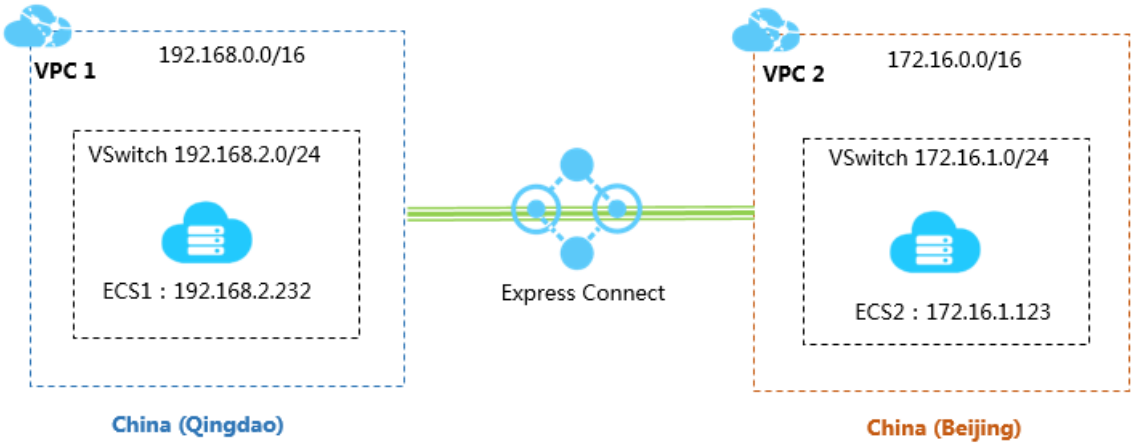
**Compared with Internet connections, Express Connect offers higher performance and security for communication between VPCs. The following table describes the differences.**

| Item | Connect VPCs through Internet connections | Connect VPCs through Express Connect |
| --- | --- | --- |
| Network performance and availability | When network traffic is transferred over a long distance, a low network latency and a low packet loss rate cannot be guaranteed. | Connections are built based on the infrastructure provided by Apsara Stack, which offers higher quality and availability. |
| Costs | You are charged a high Internet bandwidth fee or data transfer fee. | The cost for cross-region communication is minimized.<br>You are not charged for connecting VPCs that belong to the same region. |
| Security | Network traffic is transferred over the Internet. Therefore, data breaches may occur. | Networks are virtualized on Apsara Stack and isolated from each other to ensure high security. |

# 21.1.3. Architecture

Virtual border routers (VBRs) are virtualization of Express Connect circuits that are isolated by Apsara Stack. VBRs use the Layer 3 overlay and switch virtualization technologies in the Software Defined Network (SDN) architecture. Mainstream tunneling technologies are used to encapsulate data packets. Data packets from a data center are encapsulated on the switch. Then, the router between the data center and the connected virtual private cloud (VPC) encapsulates the data packets again before it forwards the packets to the VPC through a connection over an Express Connect circuit.

The following figure shows the architecture for using Express Connect to connect VPCs.

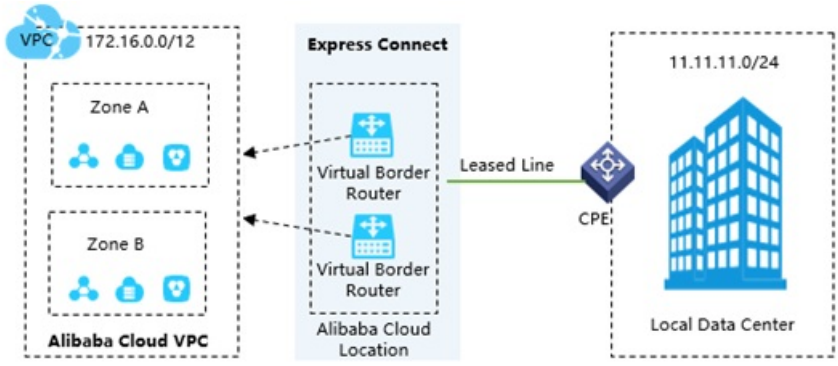



# 21.1.4. Scenarios

Express Connect enables reliable, secure, and high-speed communication between data centers and virtual private clouds (VPCs). You can use Express Connect to enable communication between networks with different architectures. The following section describes the scenarios in which you can use Express Connect.

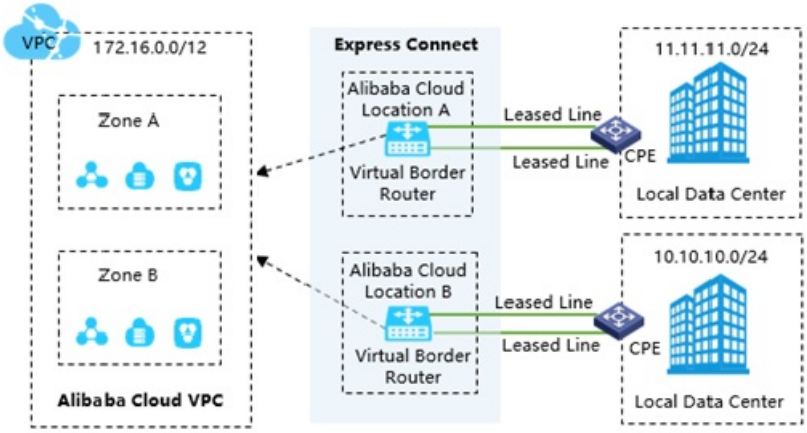
## Scenario 1: Multi-zone disaster recovery and high-availability network architecture for large and medium-sized enterprises

You can establish multiple connections over Express Connect circuits between different access points and a VPC to ensure high availability for your business-critical services. This prevents service downtime caused by severed fiber cables, device malfunctions, or access point errors.



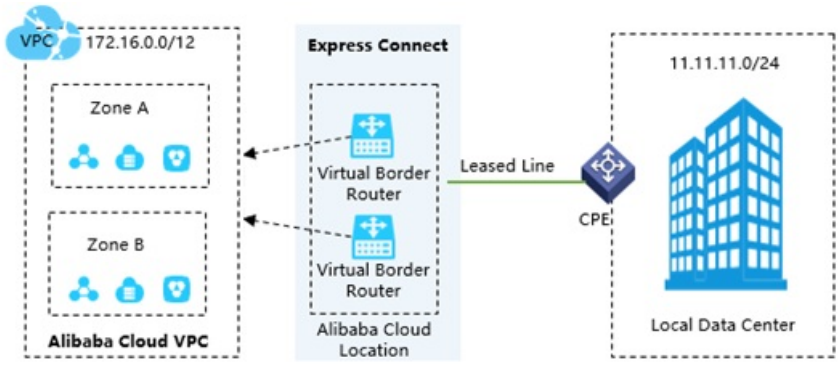

## Scenario 2: High-elasticity and high-availability network architecture for large enterprises

In scenarios where your business grows rapidly and the data center cannot meet the requirements for further development, you can deploy your workloads on the cloud. In addition, Express Connect supports equal-cost multi-path (ECMP) routing. You can configure ECMP to optimize bandwidth usage through multiple connections over Express Connect circuits and transfer network traffic at the Tbit/s level. This prevents service downtime caused by malfunctions of devices, Express Connect circuits, and access points.




## Scenario 3: Simplified network architecture for non-critical services

In scenarios that do not require high elasticity or high availability, Express Connect allows you to build a simplified network. For example, to run tests in a staging environment on Apsara Stack, we recommend that you use Express Connect to establish a private connection between your data center and Apsara Stack. This ensures the security and reliability of communication between your data center and Apsara Stack.




# 21.1.5. Terms

This topic describes the terms used in Express Connect.

The following table describes the terms that you must take note of when you use Express Connect circuits to connect to Apsara Stack.

| Term | Description |
| --- | --- |
| Express Connect | An Apsara Stack service that allows you to build private, flexible, reliable, and secure connections between VPCs and data centers. |

| Term | Description |
| --- | --- |
| Virtual Private Cloud (VPC) | A private and logically isolated network deployed on Apsara Stack. You can create and manage VPCs based on your business requirements. You can create and manage your cloud instances, such as Elastic Compute Service (ECS) instances, ApsaraDB instances, and Server Load Balancer (SLB) instances in VPCs. |
| Virtual Border Router (VBR) | A router between the customer-premises equipment (CPE) and a VPC. A VBR is used to exchange data between a data center and a VPC. |
| vRouter | Connects vSwitches in a VPC. A vRouter also serves as a gateway to connect a VPC and other networks. |
| Cloud Enterprise Network (CEN) | Allows you to create private connections between VPCs or between VPCs and data centers. CEN supports automatic route distribution and learning to accelerate network convergence. You can use CEN to create secure and reliable connections and build a global network that offers enterprise-level communication capabilities. |
| Express Connect circuits | Leased from connectivity providers. Express Connect circuits are used to connect to the access points of Apsara Stack. |
| Access points | A geographical location where an Express Connect circuit is connected to Apsara Stack. Two access devices are deployed in each access point. One or more access points are deployed in each region. You can connect a data center to a VPC through one of the access points. |
| Border Gateway Protocol (BGP) | A dynamic routing protocol based on Transmission Control Protocol (TCP). BGP is used to exchange routing information and network accessibility information in different autonomous systems. |
| Equal-cost multi-path (ECMP) routing | You can connect an access device to multiple Express Connect circuits, associate a VBR with the Express Connect circuits, and configure ECMP to make full use of the bandwidth of the connections. |
| Dedicated connections over Express Connect circuits | A connection that uses a dedicated Express Connect circuit, which is leased from a connectivity provider to connect a data center to an Apsara Stack access point. |

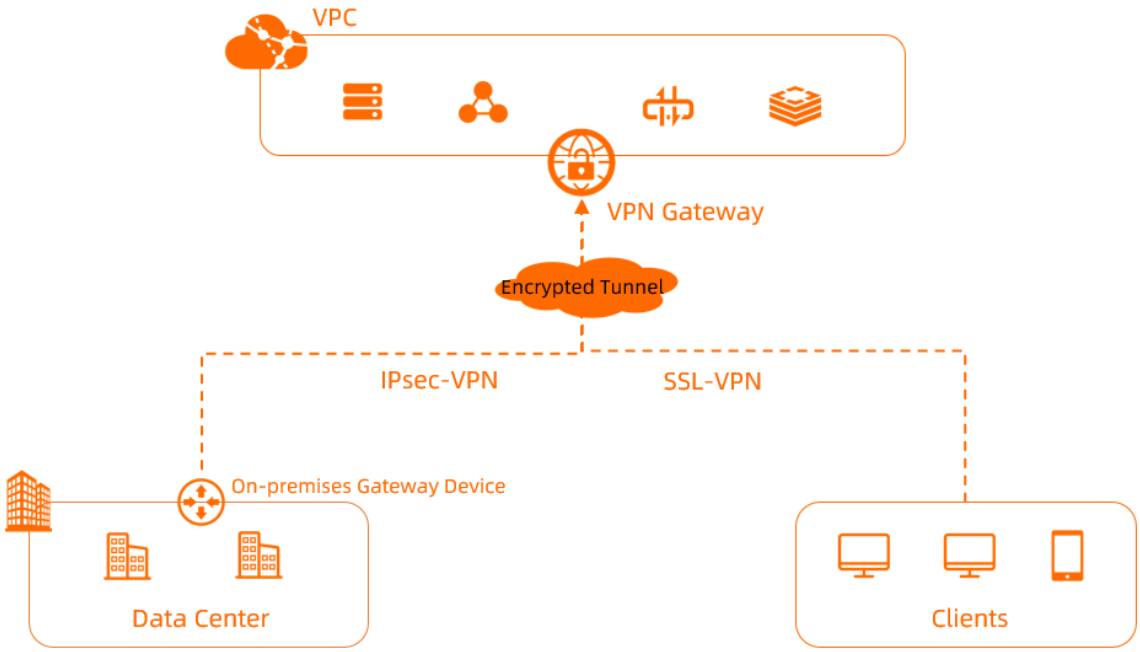
# 22.VPN Gateway

## 22.1. Product Introduction

### 22.1.1. What is VPN Gateway?

VPN Gateway is an Internet-based service that securely and reliably connects enterprise data centers, office networks, and Internet terminals to virtual private clouds (VPCs) through encrypted tunnels.

> **Note** To comply with the relevant national regulations and policies, Alibaba Cloud VPN Gateway does not provide Internet access services.




#### Features

VPN Gateway supports both IPsec-VPN connections and SSL-VPN connections.

- IPsec-VPN

  IPsec-VPN connects networks based on routes. It facilitates the configuration and maintenance of VPN policies, and provides flexible traffic routing methods.

  You can use IPsec-VPN to connect a data center to a VPC or connect two VPCs. IPsec-VPN supports the IKEv1 and IKEv2 protocols. All on-premises gateway devices that support these two protocols can connect to VPN gateways on Alibaba Cloud.

- SSL-VPN

SSL-VPN is based on OpenVPN. After you deploy the required resources, you can load the SSL client certificate on your client and initiate an SSL-VPN connection between the client and a VPC. This way, your client can access applications and services in the VPC.

### Benefits

- **Security**

  VPN Gateway uses the IKE and IPsec protocols in data transmission to ensure data security.

- **Stability**

  VPN Gateway adopts the hot-standby architecture to implement failover within a few seconds, session persistence, and zero service downtime.

- **Ease of use**

  VPN gateway is ready-to-use and its configurations immediately take effect. You can deploy VPN gateways in a fast manner.

- **Cost savings**

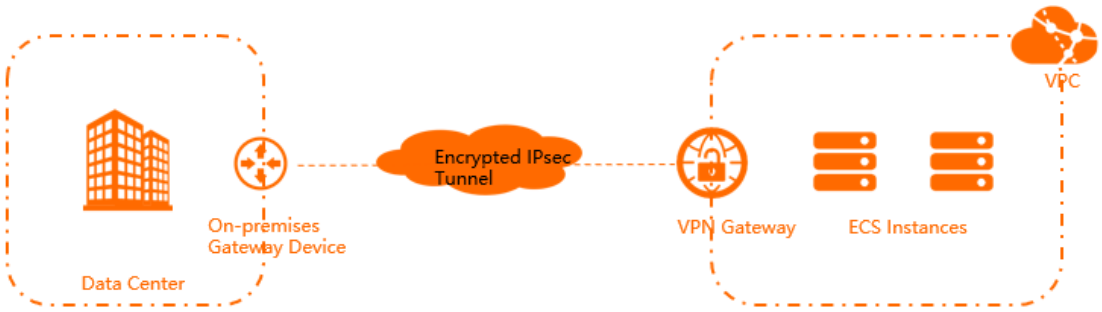
  The encrypted and Internet-based connections provided by VPN Gateway are more cost-effective than Express Connect circuits.

## 22.1.2. Scenarios

VPN Gateway is an Internet-based service that securely and reliably connects enterprise data centers, office networks, and Internet terminals to virtual private clouds (VPCs) of Alibaba Cloud through encrypted channels. This topic describes the common scenarios of VPN Gateway.

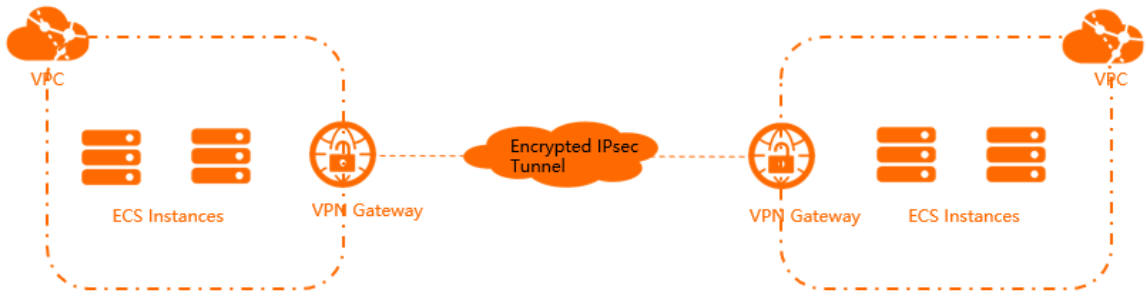
### Connect a data center to a VPC

You can use IPsec-VPN to connect a data center to a VPC and build a hybrid cloud.




### Connect two VPCs

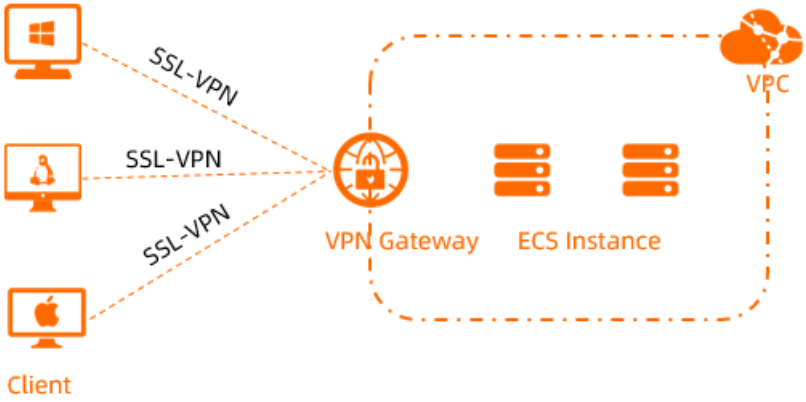
You can use IPsec-VPN to connect two VPCs. This way, cloud resources can be shared across the VPCs.




## Connect a client to a VPC

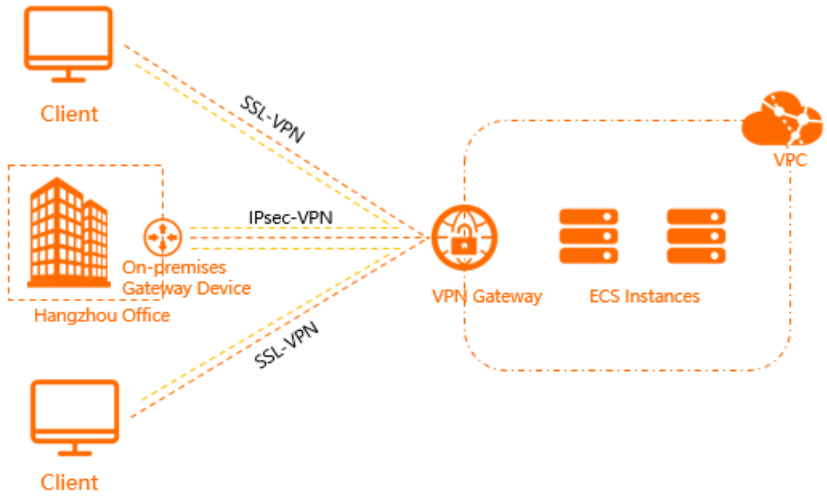
You can use SSL-VPN to connect a client to a VPC. This way, your client can securely connect to a VPC over the Internet regardless of time and location to meet your telecommuting requirements.

You can initiate an SSL-VPN connection from clients that run Windows, Linux, macOS, iOS, or Android.




## Connect a client to an office network

You can use IPsec-VPN together with SSL-VPN to connect a client and an office network to a VPC. This way, the client and the office network can access the VPC, and the client and the office network can communicate with each other.




# 22.1.3. Limits

This topic describes the limits imposed on VPN gateways and how to request a quota increase.

## Limits on instances

| Item | Limit |
| --- | --- |
| Maximum number of VPN gateways that you can create with each Alibaba Cloud account | 30<br>Note This quota is determined only by the number of Alibaba Cloud accounts and is irrelevant to regions or VPCs.<br>For example, for each Alibaba Cloud account:<br>• You can create at most 30 VPN gateways for one VPC in one region.<br>• You can create at most 30 VPN gateways for multiple VPCs in multiple regions. |
| Maximum number of policy-based route entries supported by each VPN gateway | 20 |
| Maximum number of destination-based route entries supported by each VPN gateway | 20 |

## Limits on customer gateways

| Item | Limit |
| --- | --- |
| Maximum number of customer gateways that you can create in each region | 100 |

## Limits on IPsec-VPN connections

| Item | Limit |
| --- | --- |
| Maximum number of IPsec-VPN connections that can be created on each VPN gateway | 10 |
| Maximum number of source CIDR blocks that can be added to each IPsec-VPN connection | 5 |
| Maximum number of destination CIDR blocks that can be added to each IPsec-VPN connection | 5 |

## SSL-VPN connections

| Item | Limit |
| --- | --- |
| Maximum number of SSL client certificates that you can create with each Alibaba Cloud account | 50 |
| Maximum number of SSL servers that can be associated with each VPN gateway | 1 |
| Maximum number of source CIDR blocks that can be added to each SSL server | 5 |
| Maximum number of destination CIDR blocks that can be added to each 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 |
| Validity period of an SSL client certificate | Three years |

# 23.Apsara Stack DNS

# 23.1. Product Introduction

## 23.1.1. What is Apsara Stack DNS?

Apsara Stack DNS is a service that is encapsulated based on the Domain Name System (DNS) protocol and runs on Apsara Stack to resolve domain names over internal networks, such as virtual private clouds (VPCs), networks of self-managed 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, self-managed business applications, business systems in 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, networks of self-managed data centers, and the classic network. You can perform the following operations by using Apsara Stack DNS in these internal networks:

- Access other Elastic Compute Service (ECS) instances deployed in the same VPC.
- Access other cloud service instances on Apsara Stack.
- Access the custom business systems of your enterprise.
- Access services over the Internet.
- Use the GSLB service to implement multi-active solutions and disaster recovery, such as local active-active, active zone-redundancy, remote active-active, active geo-redundancy, and geo-disaster recovery.
- Connect to Apsara Stack DNS with your own DNS servers over a leased line.

## 23.1.2. Edition comparison

This topic describes the differences between Apsara Stack DNS editions.

> **Note** In Apsara Stack Enterprise Edition, four DNS editions are supported: DNS Lightweight Basic Edition, DNS Basic Edition, DNS Standard Edition, and Internal GTM Standard Edition. In Apsara Stack Agility Edition, two DNS editions are supported: DNS Basic Edition and Internal GTM Standard Edition. In GTM for multi-cloud, only one DNS edition is supported: GTM In-cloud.

| Feature category | Feature | DNS Lightweight Basic Edition | DNS Basic Edition | DNS Standard Edition | Internal GTM Standard Edition |
|---|---|---|---|---|---|
| | Global basic DNS resolution | Supported | Supported | Supported | Supported |

| Feature category | Feature | DNS Lightweight Basic Edition | DNS Basic Edition | DNS Standard Edition | Internal GTM Standard Edition |
|---|---|---|---|---|---|
| Global domain name in Apsara Stack | Global load balancing (weight) | Supported | Supported | Supported | Supported |
| | Global domain name forwarding | Supported | Supported | Supported | Supported |
| | Global default forwarding | Supported | Supported | Supported | Supported |
| | Internet recursive resolution | Supported | Supported | Supported | Supported |
| Private domain name for VPC | Tenant-specific basic DNS resolution | Not supported | Not supported | Supported | Supported |
| | Tenant-specific load balancing (weight) | Not supported | Not supported | Supported | Supported |
| | Tenant-specific domain name forwarding | Not supported | Not supported | Supported | Supported |
| | Tenant-specific default forwarding | Not supported | Not supported | Supported | Supported |
| | VPC association and disassociation | Not supported | Not supported | Supported | Supported |
| Scheduling instance | Scheduling instance management (multicloud) | Not supported | Not supported | Not supported | Supported |
| | Address pool management (multicloud) | Not supported | Not supported | Not supported | Supported |

| Feature category | Feature | DNS Lightweight Basic Edition | DNS Basic Edition | DNS Standard Edition | Internal GTM Standard Edition |
|---|---|---|---|---|---|
| Global line | Scheduling domain management | Not supported | Supported | Supported | Supported |
| Private line | Scheduling domain management | Not supported | Not supported | Supported | Supported |
| Node | Synchronization cluster management (multicloud) | Not supported | Supported | Supported | Supported |
| Global data synchronization (multicloud) | Not supported | Supported | Supported | Supported | |
| Log | Log management (multicloud) | Not supported | Not supported | Not supported | Supported |
| Cross-cloud shared domain name | Global basic DNS resolution (multicloud) | Not supported | Not supported | Not supported | Supported |
| ISP line | ISP line management (multicloud) | Not supported | Not supported | Not supported | Supported |
| Other features | Independent physical machine deployment | Not required | Required | Required | Required. The physical machines of this edition must be deployed with the physical machines of DNS Basic Edition or DNS Standard Edition. |
| | Web-based graphical user interface (GUI) | Supported | Supported | Supported | Supported |

| Feature category | Feature | DNS Lightweight Basic Edition | DNS Basic Edition | DNS Standard Edition | Internal GTM Standard Edition |
|---|---|---|---|---|---|
| | Update to DNS Basic Edition | Supported. New physical machines must be deployed. | Not supported | Not supported | Not supported |
| | Update to DNS Standard Edition | Supported. New physical machines must be deployed. | Supported | Not supported | Not supported |

# 23.1.3. 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:

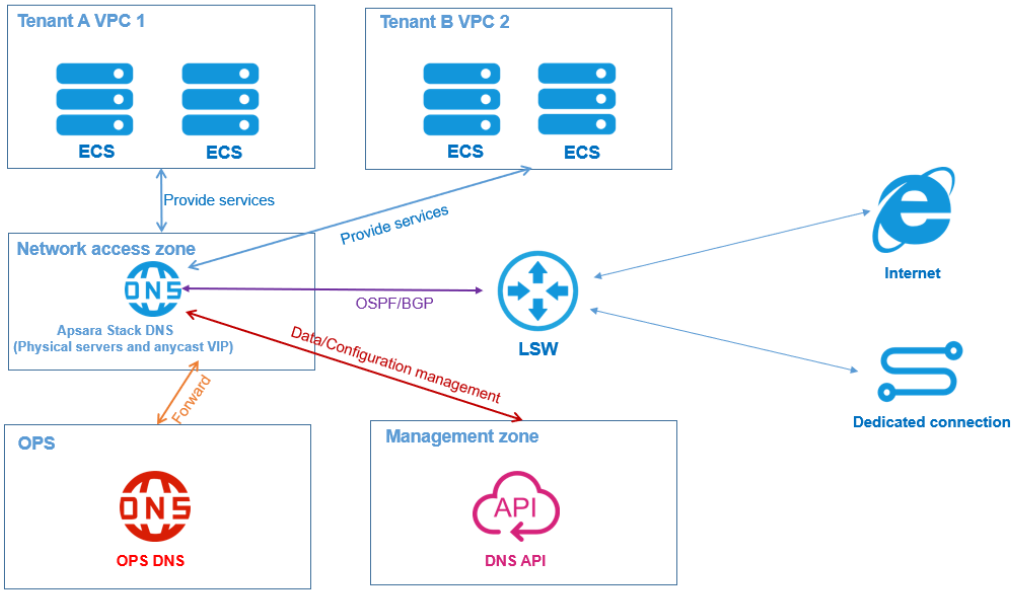
- Allows you to add multiple A, AAAA, or CNAME records at a time. DNS servers respond to DNS queries based on the weight of each record type to achieve load balancing.
- Supports scheduling line management. You can customize lines and their priorities to allow clients to access the nearest servers and 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.

## 23.1.4. 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.

# 23.1.5. 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 failback 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.

## 23.1.6. 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.

## 23.1.7. Limits

This topic describes limits on Apsara Stack DNS clusters.

DNS Basic Edition

| 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 |

| 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 |

### 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 |
| 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.

# 23.1.8. 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.

# 24.Log Service

# 24.1. Product Introduction

## 24.1.1. What is Log Service?

Log Service is an all-in-one service that is developed by Alibaba Group based on extensive big data analytics scenarios. You can use Log Service to collect, consume, query, and analyze log data without the need to invest in in-house data collection or processing systems. Log Service helps improve the O&M efficiency and allows you to process large amounts of data.

Log Service provides the following features:

- Log collection: Log Service allows you to collect events, binary logs, and text logs in real time by using multiple methods, such as Logtail and JavaScript.
- Query and analysis: Log Service allows you to query collected log data in real time and analyze the collected log data. Log Service also allows you to view analysis results in charts and on dashboards.
- Real-time consumption: Log Service provides real-time consumption interfaces that log consumers can use to consume log data.

## 24.1.2. Benefits

This topic describes the benefits of Log Service.

### Fully managed services

- Log Service is easy to access and easy to use.
- LogHub provides all features of Kafka, provides monitoring and alerting data, and supports auto scaling (by petabytes per day).
- LogSearch/Analytics provides the saved search feature, and allows you to view log data on dashboards and configure alerts.
- Log Service provides more than 30 access methods to connect to open source software, such as Storm and Spark Streaming.

### Inclusive ecosystem

- LogHub supports more than 30 types of data sources, including embedded devices, web pages, servers, and programs. LogHub can connect to different consumption systems, such as Storm and Spark Streaming.
- LogSearch/Analytics supports complete query and analysis syntax and is compatible with SQL-92 syntax. Log Service can be connected to Grafana by using Java Database Connectivity (JDBC).

### Real-time response

- LogHub: Data can be immediately consumed after the data is written to Log Service. Logtail is used as an agent to collect and send data to Log Service in real time.

- LogSearch/Analytics: Data can be queried 3 seconds after the data is written to Log Service. If you specify multiple conditions to query data from hundred billions of data records, the result can be returned in seconds.

### High throughput

Log Service provides high throughput after Log Service software is optimized. If Log Service is deployed on a single server, the throughput performance can reach the following levels:

- Data write: Raw logs can be written to Log Service at a speed that exceeds 400 MB/s, and indexed logs can be written to Log Service at a speed that exceeds 150 MB/s.
- Real-time consumption: Data can be consumed in real time at a speed that exceeds 400 MB/s.
- QPS: The queries per second (QPS) can exceed 10,000.

> **Note** The throughput performance varies based on the hardware of a server.

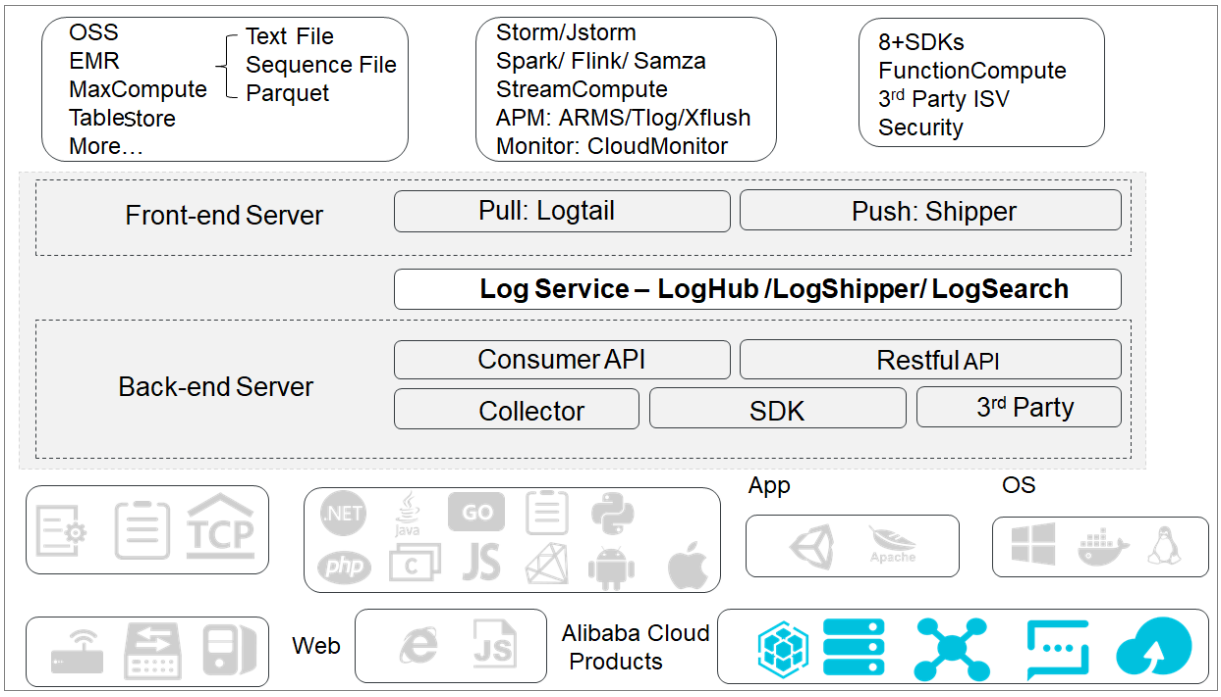
### Scalable

Log Service provides auto scaling capabilities to process petabytes of data.

## 24.1.3. 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.

# 24.1.4. Features

## 24.1.4.1. Core features

This topic describes the following core features of Log Service: real-time log collection and consumption, and real-time log query and analysis.

### Real-time log collection and consumption (LogHub)

LogHub allows you to collect logs without data loss by using various collection methods, such as Logtail, SDKs, web pages, protocols, and API operations. LogHub allows you to consume logs by using methods such as SDKs, Storm Spout, and Spark Client. LogHub also allows you to collect logs in various formats in real time and consume the logs that are collected. You can use LogHub to streamline the collection process and consumption process of logs across devices and sources.

Features:

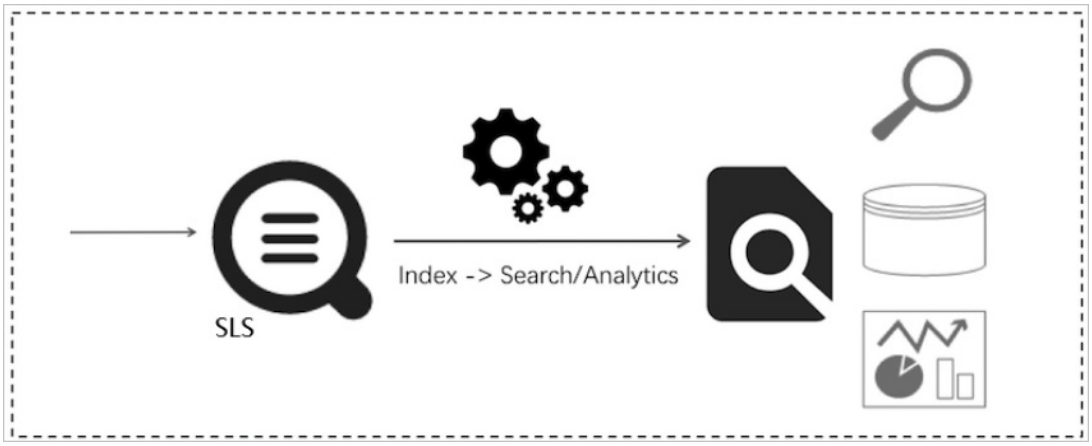
- LogHub collects real-time log data from different sources, such as Elastic Compute Service (ECS) instances, containers, mobile terminals, open source software, and JavaScript. The log data includes metrics, events, binary logs, text logs, and clicks. LogHub can collect data at a speed of 100 MB/s per core by using Logtail and Log Service SDK for C or C++.
- LogHub provides real-time consumption interfaces to connect Log Service to real-time computing engines and services.

### Real-time log query and analysis (Search/Analytics)

Log Service allows you to index, query, and analyze the log data that is collected to Log Service in real time. Log Service can generate dynamic reports based on the query and analysis results. Log Service allows you to analyze data in multiple scenarios and view the analysis results.

- Query: You can query data by using keywords, fuzzy match, and context. You can also query data within a specified range.
- Statistics: After you query data, you can use various methods such as SQL aggregate functions to analyze the data.
- Visualization: You can view query and analysis results on dashboards and export the results as reports.
- Interconnection: Log Service can be connected to Grafana by using Java Database Connectivity (JDBC). SQL-92 is supported.

  Real-time log query and analysis




# 24.1.4.2. Other features

## 24.1.4.2.1. 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 |
| --- | --- | --- |
| 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. |

| Field | Description | Format |
| --- | --- | --- |
| 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: `_time`, `source`, `topic__`, `partition_time_`, `extract_others`, and `__extract_others__`. 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. |
| 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.

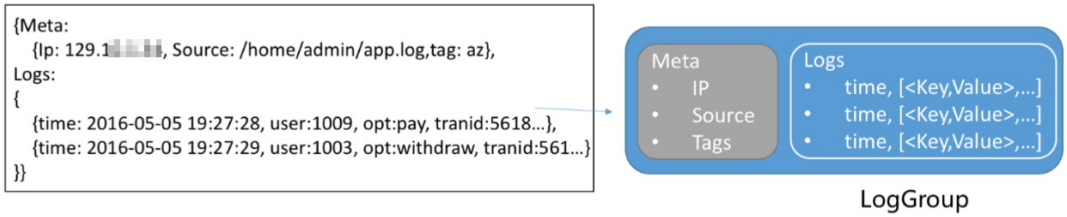
| Key | Value |
| --- | --- |
| ip | 10.1.168.193 |
| method | GET |
| status | 200 |
| length | 5 |
| ref_url | N/A |
| browser | Mozilla/5.0 (X11; Linux i686 on x86_64; rv:10.0.2) Gecko/20100101 Firefox/10.0.2 |

### 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




## 24.1.4.2.2. 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.

## 24.1.4.2.3. 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.

# 24.1.4.2.4. 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, ffffffffffffffffffffffffffffffff).

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) |

| Shard ID | MD5 hash key range |
| --- | --- |
| 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.

## 24.1.4.2.5. 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.

# 24.1.5. 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

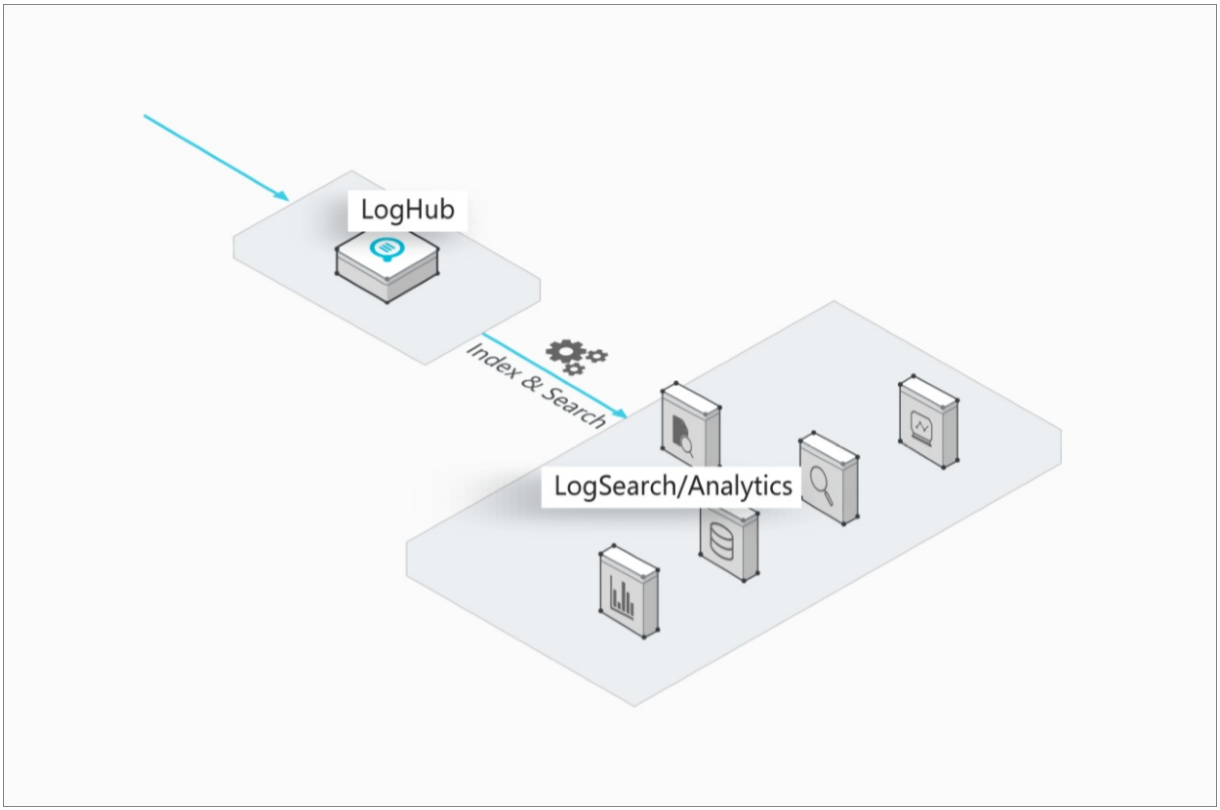
You can connect LogHub to multiple real-time computing engines and services. LogHub can monitor the processing progress and generate alerts based on the monitoring results. You can also use SDKs or API operations to consume logs.

- 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.




# 24.1.6. 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.<br>• 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.<br>• 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. |

# 24.1.7. 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.

# 25.API Gateway

# 25.1. Product Introduction

## 25.1.1. What is API Gateway?

API Gateway is an API hosting service. It provides a full range of lifecycle management features, including API design, development, testing, publishing, O&M and monitoring, security control, and publishing. API Gateway has the following benefits:

- Helps build an API-centric system architecture and meets the needs in different scenarios such as the introduction of new technologies, system integration, and Business Mid-end.
- Provides multiple security mechanisms to secure APIs and reduce the risks arising from APIs. These mechanisms include protection against replay attacks, request encryption, identity authentication, permission management, and throttling.
- Helps automatically generate SDK references and API references. This improves the efficiency of API management and iteration.
- Improves the reusability of different capabilities. This accelerates business innovation inside enterprises.

### Scenarios

#### I. API management hub for Business Mid-end

API Gateway can manage APIs of various systems in a centralized manner by leveraging its interconnection and integration capabilities. Centralized API management, including throttling, permission management, and monitoring, facilitates O&M and allows you to configure a single API that can be called by multiple systems. This better improves operational efficiency.

#### II. API compatibility with multiple types of terminals

As mobile networks and IoT develop, APIs need to support more types of terminals to be suitable in more business scenarios. However, this increases system complexity.

1. In API Gateway, enterprises can manage and maintain APIs in a single service system and adapt APIs to different types of terminals, such as apps, devices, and web clients, only by changing API definitions.
2. Enterprises can develop and manage a single API for multiple scenarios, multiple types of terminals and users, and multi-tier services. This reduces the costs and complexity of O&M.

#### III. System integration

1. API Gateway helps standardize APIs of different systems. This way, you can integrate systems with standard APIs.

2. API Gateway helps integrate and manage resources with efficiency and prevents resource redundancy and waste caused by fast development. This way, you can focus on business development.

# 25.1.2. Architecture

API Gateway is an API hosting service. It provides a full range of lifecycle management features to help build an API-centric system architecture. The lifecycle management features include API design, development, testing, publishing, O&M and monitoring, security control, and unpublishing.

API Gateway consists of three components:

- Gateway: The gateway component is the core system that implements all the business logic of API Gateway. The gateway component supports access from all clients over multiple protocols, including HTTP, HTTPS, and WebSocket. The gateway component manages client connections, throttles API requests, and implements IP address-based access control. The gateway component loads user-defined APIs into the memory, processes requests from clients based on API definitions, calls backend APIs, and returns backend responses to clients.
- API Gateway API: The API Gateway API consists of a group of standard management operations that are used to manage API definitions. You can use the API Gateway API to manage groups, metadata, and authorization for APIs. When the API Gateway API receives an API change request, it synchronizes the change to all gateway services. System administrators can use the management operations to manage the APIs that are running in API Gateway in real time. System administrators can manage their own APIs in the API Gateway console in real time. They can also call the management operations in their own management systems to manage their own APIs.
- API Gateway console: The API Gateway console implements all features of API Gateway. System administrators can manage their own APIs in the console in real time. The API Gateway console calls the operations of the API Gateway API to provide web-based operations.

# 25.1.3. Benefits

- Less workload

After you create and configure APIs in API Gateway, API Gateway performs all the other API management operations, such as documentation maintenance and version management for APIs, and SDK maintenance. This significantly reduces routine maintenance costs.

- High performance

API Gateway supports efficient access over HTTP/2 and maintains persistent connections by supporting the binary protocol WebSocket. This improves the performance of the connections between clients and API Gateway. API Gateway adopts distributed deployment and automatically scales out to handle a large number of API requests with low latency. API Gateway offers reliable and efficient features for your backend services.

- Stability

In 2016, API Gateway was released for commercial use on Alibaba Cloud public cloud. API Gateway has stood the test on both Alibaba Cloud public cloud and Apsara Stack over the years. API Gateway can maintain stable operation even in special cases where oversized messages are received, or the backend service is unstable and does not respond at the earliest opportunity.

- Security

API Gateway implements SSL encryption in the full link of communication to protect all data against eavesdropping during transmission. API Gateway implements signature verification in the full link of communication to prevent data tampering during transmission. To ensure that your services are secure, stable, and controllable, API Gateway also provides a set of API security features. The features include strict permission management, replay attack prevention, parameter cleansing, IP address-based access control, precise throttling, and integration with Web Application Firewall (WAF) of Alibaba Cloud.

# 25.1.4. Features

## API lifecycle management

- Manages APIs throughout their entire lifecycle, including publishing, testing, and unpublishing APIs.
- Supports API maintenance features such as routine management, version management, and quick rollback.
- Allows APIs in different environments to be accessed by using different domain names or headers.
- Supports the API diff feature. When you release a new version of an API, you can check the differences between the new version and an earlier version of the API.

## Comprehensive security protection

- Supports multiple authentication methods, including anonymous access, simple authentication, signature authentication, and JSON Web Token (JWT) authentication.
- Supports HTTPS and SSL encryption.
- Provides multiple security mechanisms to prevent injections, replay attacks, and tampering.
- Supports backend signature authentication. Your backend service can authenticate API Gateway based on signatures.

## Flexible access control

- Manages the API call permissions of APPs.
- Allows only authorized APPs to call APIs.
- Allows API owners to authorize APPs to call APIs.

## Various plug-in features

- Allows you to use various plug-ins that can be bound or unbound to expand the features of APIs.
- 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. The value ranges, enumerated values, and regular expressions in values can be verified. API Gateway denies the requests with invalid parameter types or values. This reduces backend resources that are wasted on invalid requests and significantly reduces the costs of processing requests for the backend services.

### Data conversion

Allows you to configure rules for mappings between the frontend data and backend data.

- Supports data conversion for frontend requests.

### Integration

Support big data platforms as backend services. This allows you to use APIs to provide data services.

### 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 to track the status of each API in real time.
- Delivers the logs about HTTP requests and responses of APIs to Log Service for full log query and analysis.

## 25.1.5. Terms

Before you use API Gateway, familiarize yourself with the following terms.

### APP

An APP is the identity of an API caller. To call an API, you must create an APP first.

### AppKey and AppSecret

Each APP has a key pair that consists of an AppKey and an AppSecret. This key pair is encrypted and attached to a request as the signature.

### encrypted signature

An encrypted signature is a signature string that is included in an API request and used by API Gateway for identity authentication.

### authorization

Authorization means that the API owner grants an APP the permissions to call an API. Only authorized APPs can call APIs.

### API lifecycle

The API owner manages an API in stages, including the creation, test, publish, unpublish, and version change stages.

## API definition

The definition of an API operation is a set of configurations that are completed by the API owner, including the backend service, request format, parameter mapping rules, and response format.

## parameter mapping

To ensure that each API request can be converted to the specified format that can be received by the backend service of an API operation, API Gateway allows the API owner to configure parameter mapping rules.

## parameter verification

Parameter verification means checking whether parameters in an API request are valid based on a set of rules that the API owner defines. API Gateway denies invalid requests.

## constant parameter

A constant parameter is a parameter that is not specified in an API request but is still received together with the request by the backend service of the API.

## system parameter

A system parameter is a parameter that you want API Gateway to attach to each API request before API Gateway routes the request to the backend service. For example, CaClientIp is a system parameter that specifies the IP address of a client.

## API group

In API Gateway, APIs are managed in API groups. 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 calls.

## independent domain

An independent domain is a domain name that you bind to an API group when you publish an API in the group. You must access the independent domain to call an API.

## plug-in

A plug-in is an extension that can be bound or unbound to implement API features.

## signature key

A signature key is created by the API owner 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 owner can configure a throttling policy to limit the maximum number of requests for an API and the maximum number of API requests that a user or an APP can initiate. The throttling granularity can be day, hour, or minute.

# 26.Message Queue for Apache RocketMQ

## 26.1. Product Introduction

### 26.1.1. What is Message Queue for Apache RocketMQ?

Message Queue for Apache RocketMQ is a distributed messaging middleware that is developed based on Apache RocketMQ. Message Queue for Apache RocketMQ features low latency, high concurrency, high availability, and high reliability.

Message Queue for Apache RocketMQ provides a complete set of cloud messaging services based on the technologies that are used for building highly available and distributed clusters. The messaging services include message subscription and publishing, message tracing, scheduled and delayed messages, and resource statistics. Message Queue for Apache RocketMQ is used as a core service in an enterprise-grade Internet architecture. Message Queue for Apache RocketMQ provides asynchronous decoupling and peak-load shifting capabilities for distributed application systems. It also supports various features for Internet applications, including accumulation of large numbers of messages, high throughput, and reliable message consumption retries. Message Queue for Apache RocketMQ is one of the core cloud services that are used to support the Double 11 Shopping Festival.

Message Queue for Apache RocketMQ supports connections over TCP and HTTP and supports multiple programming languages such as Java, C++, and .NET. This allows you to connect applications that are developed in different programming languages to Message Queue for Apache RocketMQ.

### 26.1.2. Updates

This topic describes the updates of Message Queue for Apache RocketMQ from V3.8.0 to V3.8.1 to help you get started with the updated version.

#### Optimization of resource isolation by instance

Message Queue for Apache RocketMQ provides instances for multi-tenancy isolation. Each user can purchase multiple instances and logically isolate them from each other.

To ensure the compatibility with the existing resources of existing users, Message Queue for Apache RocketMQ provides the following types of instances and namespaces:

- Default instances, which are compatible with the existing resources of existing users
  - This type of instance has no separate namespace. Resource names must be globally unique within and across all instances.
  - By default, an instance without a namespace is automatically generated for the existing resources of each existing user. If no existing resources are available, you can create at most one instance without a namespace.

- You can configure the endpoint, which can be obtained from the **Instances** page in the Message Queue for Apache RocketMQ console.

```
// Recommended configuration:
properties.put(PropertyKeyConst.NAMESRV_ADDR, "xxxx");
// Compatible configuration, which is not recommended. We recommend that you update this configuration to the recommended configuration:
properties.put(PropertyKeyConst.ONSAddr, "xxxx");
```

- New instances
  - A new instance has a separate namespace. Resource names must be unique within an instance but can be the same across different instances.
  - You can configure the endpoint, which can be obtained from the **Instances** page in the Message Queue for Apache RocketMQ console.

```
// Recommended configuration:
properties.put(PropertyKeyConst.NAMESRV_ADDR, "xxx");
```

  - A RocketMQ client must be updated to the following latest versions for different programming languages:
    - Java: V1.8.7.1.Final
    - C and C++: V2.0.0
    - .NET: V1.1.3

## Optimization of resource application

Previously, Message Queue for Apache RocketMQ resources consisted of topics, producer IDs, and consumer IDs. Each two of the resources have a many-to-many relationship, which was difficult to comprehend. Each time you created a topic, you must associate the topic with a producer ID and a consumer ID. This process was too complex for medium- and large-sized enterprise customers.

To optimize user experience and help new users get started, the resource application process has been simplified.

The resource application process has been optimized in the following aspects:

- Topic management, which is unchanged
  - You need to apply for a topic. A topic is used to classify messages. It is the primary classifier.
- Group management
  - You do not need to apply for a producer ID. Producer IDs and consumer IDs are integrated into group IDs. In the Message Queue for Apache RocketMQ console, the Producers module has been removed. The Producers and Consumers modules have been integrated into the Groups module.
  - You do not need to associate a producer ID or consumer ID with a topic. Instead, you need only to apply for a group ID and associate it with a topic in the code.

- Compatibility:
  - The list of producer IDs is no longer displayed. This does not affect the current services.
  - The consumer IDs that start with CID- or CID_ and that you have applied for can still be used and can be set in the PropertyKeyConst.ConsumerId or PropertyKeyConst.GROUP_ID parameter of the code.

- Sample code

  > Note
  > - We recommend that you update a RocketMQ client to the following latest versions for different programming languages:
  >   - Java: V1.8.7.1.Final
  >   - C and C++: V2.0.0
  >   - .NET: V1.1.3
  > - Existing producer IDs or consumer IDs can still be used and do not affect the current services. However, we recommend that you update your instance configuration to the recommended configuration.

  - Recommended configuration: Integrate producer IDs and consumer IDs into group IDs.

    ```
    // Set the PropertyKeyConst.GROUP_ID parameter. The original PropertyKeyConst.ProducerId and Prop
    ertyKeyConst.ConsumerId parameters are deprecated.
    properties.put(PropertyKeyConst.GROUP_ID, "The original CID-XXX or the GID-XXX");
    ```

  - Compatible configuration: Use a producer ID to identify a producer and a consumer ID to identify a consumer.

    ```
    // When you create a producer, you must set the PropertyKeyConst.ProducerId parameter.
    properties.put(PropertyKeyConst.ProducerId, "The original PID-XXX or the GID-XXX");
    // When you create a consumer, you must set the PropertyKeyConst.ConsumerId parameter.
    properties.put(PropertyKeyConst.ConsumerId, "The original CID-XXX or the GID-XXX");
    ```

# 26.1.3. Benefits

This topic describes the advantages of Message Queue for Apache RocketMQ over other messaging middleware.

## Professionalism

- Message Queue for Apache RocketMQ is a professional messaging middleware in the industry. This service ensures data integrity and features a sophisticated technical system.
- Message Queue for Apache RocketMQ is developed based on open source Apache RocketMQ. Message Queue for Apache RocketMQ has won several awards both in China and abroad.
- More than 1,000 core applications within Alibaba Cloud use Message Queue for Apache RocketMQ. This service forwards hundreds of billions of messages per day. It delivers stable and reliable performance in real-life business scenarios such as the Double 11 Shopping Festival.

### High reliability

- Each message is stored in multiple replicas on disks. No message is found lost after rigorous power-off tests.
- Message Queue for Apache RocketMQ supports the accumulation of large numbers of messages. A single topic can reliably support more than 10 billion messages even when the workload of the system is high.
- By default, messages are stored for three days. Message Queue for Apache RocketMQ allows you to reset consumer offsets for messages that are produced at a point in time within three days.

### High availability

- Message Queue for Apache RocketMQ provides the active geo-redundancy feature. This feature allows you to implement fast failovers without downtime in business failure scenarios. This way, business recovery and fault recovery are decoupled, and business continuity is ensured.

### High performance

- Scaling out is supported.
- The maximum size of a single message that is supported in Message Queue for Apache RocketMQ is 4 MB. When the system calculates the size of a message, message properties are also counted.

### Support for multiple protocols

- Message Queue for Apache RocketMQ supports connections over TCP and HTTP and supports multiple programming languages such as Java, C++, and .NET.

### Independent deployment

- Message Queue for Apache RocketMQ can be independently deployed in Apsara Stack or on physical machines. Only a few machines are required to build a complete messaging system on Message Queue for Apache RocketMQ.
- Apsara Stack provides the mqadmin command set and management API operations. O&M engineers can conveniently monitor the health status of your system in real time.
- Message Queue for Apache RocketMQ can be deployed in a hybrid cloud architecture and allows you to use Express Connect circuits to connect to services in the Alibaba Cloud public cloud.

## 26.1.4. Architecture

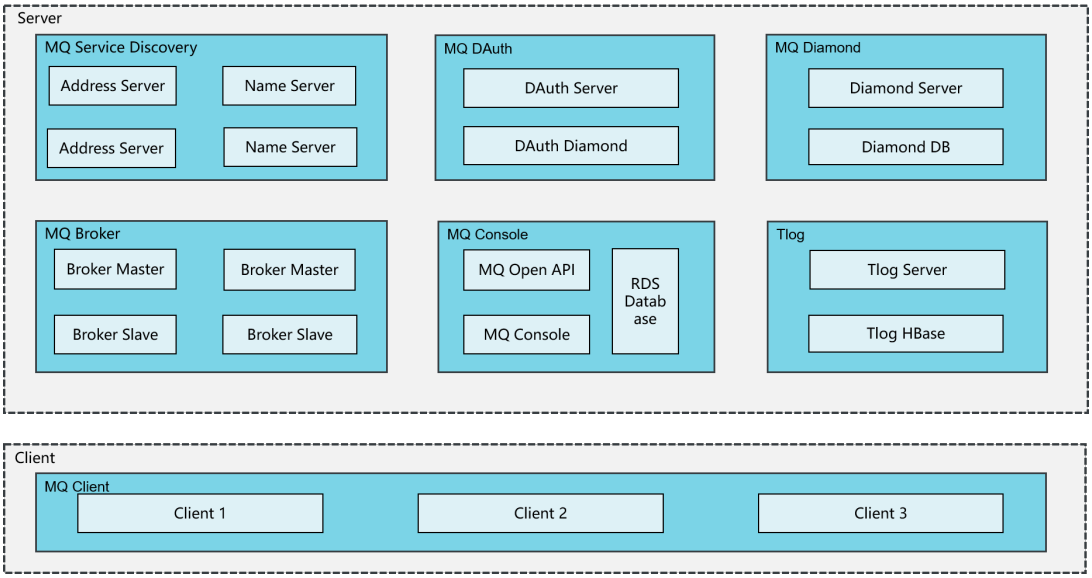
This topic describes the system architecture of Message Queue for Apache RocketMQ to help you better understand the features of each component.

### Overview

Message Queue for Apache RocketMQ consists of various components, including the RocketMQ service registry, RocketMQ control nodes, RocketMQ brokers, RocketMQ DAuth, RocketMQ Diamond, and Tlog.

RocketMQ system architecture diagram

RocketMQ system architecture diagram




| Component | Description |
| --- | --- |
| RocketMQ service registry | The RocketMQ service registry consists of Name Servers and Address Servers. A Name Server is a core component for implementing the flexible deployment and linear scaling of message queues and is responsible for registering and searching for message queues. An Address Server, also called Cai, is responsible for registering and discovering domain names for Name Servers. |
| RocketMQ broker | A RocketMQ broker is a core component for processing message queues. Multiple brokers compose a cluster and are responsible for receiving, sending, and storing messages. |
| RocketMQ console | The Message Queue for Apache RocketMQ console allows you to manage resources, detect issues at the earliest opportunity, and troubleshoot issues. For example, you can manage topics and groups, query messages, query message traces, and collect resource statistics. |
| RocketMQ API | The Message Queue for Apache RocketMQ API provides a set of API operations that you can call by using HTTP and HTTPS. This helps you use Message Queue for Apache RocketMQ and manage resources with ease. For example, you can call API operations to create topics, query group IDs, query messages, and query the status of consumers. |
| RocketMQ DAuth | RocketMQ DAuth provides unified logon and access control for message queues, including resource permission control, cross-account and RAM user access control, and resource authorization. |

| Component | Description |
| --- | --- |
| RocketMQ Diamond | RocketMQ Diamond is a configuration center that stores the configuration information of message queues, including VIP conversion rules and resource permission information. |
| Tlog | Tlog collects resource statistics and other key logs of Message Queue for Apache RocketMQ. |

# 26.1.5. Functions and features

Message Queue for Apache RocketMQ supports connections over TCP and HTTP from clients that are developed in multiple programming languages, and offers multi-dimensional management tools. Message Queue for Apache RocketMQ also provides various features to meet business requirements in different scenarios.

## Support for multiple protocols

- Message Queue for Apache RocketMQ supports connections over TCP and HTTP and provides SDKs for multiple programming languages, such as Java, C, C++, and .NET.

## Management tools

- Web console: Message Queue for Apache RocketMQ provides a web console in which you can manage topics and groups, query messages and message traces, and view resource statistics.
- API: Message Queue for Apache RocketMQ provides API operations that allow you to integrate the management tools into your console.
- mqadmin command set: Apsara Stack provides a rich set of management commands. You can run commands to manage your resources in Message Queue for Apache RocketMQ.

## Message types

- Normal message: In Message Queue for Apache RocketMQ, messages of this type do not provide special features.
- Scheduled message or delayed message: Message Queue for Apache RocketMQ allows producers to specify the delay period to wait before brokers deliver a scheduled or delayed message. The maximum delay period is 40 days.
- Transactional message: Message Queue for Apache RocketMQ provides a distributed transaction processing feature that is similar to X/Open XA to ensure the eventual consistency of transactions.
- Ordered message: Message Queue for Apache RocketMQ allows consumers to consume messages in the order in which the messages are sent.

## Feature highlights

- Large messages: The maximum size of a single message that is supported in Message Queue for Apache RocketMQ is 4 MB. When the system calculates the size of a message, message properties are also counted.
- Message query: Message Queue for Apache RocketMQ allows you to query messages by

message ID, message key, and topic.

- Message tracing: This feature records the complete trace of each message from its delivery by a producer to a RocketMQ broker and then to a consumer. This facilitates troubleshooting.
- Clustering consumption and broadcasting consumption: In clustering consumption mode, a message needs to be processed by only one of the consumers in a consumer group. In broadcasting consumption mode, Message Queue for Apache RocketMQ pushes each message to all registered consumers in a consumer group to ensure that each message is consumed by each consumer at least once.
- Consumer offset resetting: You can reset the consumption progress by time to analyze message traces or discard accumulated messages.
- Dead-letter queues: Messages that cannot be consumed as expected are stored in a special dead-letter queue for subsequent processing.
- Resource statistics: You can use this feature to collect statistics about message production and message consumption. This feature allows you to view the total number of messages that a topic receives or the transactions per second (TPS) for message production within a specified period of time. This feature also allows you to view the total number of messages that are delivered from a topic to a consumer group or the TPS for message consumption within a specified period of time.
- Active geo-redundancy: You can purchase active geo-redundancy instances in the Message Queue for Apache RocketMQ console. If you deploy Multi-Site High Availability (MSHA) for your project, your business workloads are divided into different production centers based on regions. Each production center provides external services at the same time, and data is synchronized and backed up between the production centers. When one of the production centers fails, you can use MSHA to switch the workloads to other healthy disaster recovery sites. This enables smooth business migration and fast recovery.

### Deployment in Apsara Stack

- Customization: Technical solutions and onsite technical support and training are provided.
- Flexible deployment: Message Queue for Apache RocketMQ can be independently deployed in Apsara Stack or deployed in a hybrid cloud architecture.
- O&M management: Apsara Stack supports O&M management tools such as the mqadmin command set and API operations. This way, you can integrate various management tools into the Message Queue for Apache RocketMQ console and perform unified O&M operations.

## 26.1.6. Scenarios

Message Queue for Apache RocketMQ is applicable to distributed transactions, real-time computing, Internet of Things (IoT) applications, and large-scale cache synchronization. This topic describes the application of Message Queue for Apache RocketMQ in different scenarios.

### Distributed transactions

In traditional transaction processing mode, interactions among systems are coupled into one transaction. This extends the response time and subsequently affects system availability. The introduction of distributed transactional messages creates a transaction processing process between transaction systems and Message Queue for Apache RocketMQ. The process ensures data consistency between distributed systems. Downstream business systems such as shopping carts and points are isolated from each other and concurrently process transactions.

### Real-time computing

Data that is continuously generated by a source flows to a computing engine in real time by using Message Queue for Apache RocketMQ. This achieves real-time computing. You can use the following computing engines: Spark, Storm, E-MapReduce, Application Real-Time Monitoring Service (ARMS), and Beam Runner.

### IoT applications

IoT devices connect to the cloud by using Message Queue for MQTT for bidirectional communication and data transmission. Device data is connected to a computing engine by using Message Queue for Apache RocketMQ. Analysis data or source data is efficiently written to a data store such as a time series database (TSDB), HiStore, or MaxCompute in real time.

### Large-scale cache synchronization

In business promotion activities such as the Double 11 Shopping Festival, each branch has a wide range of products, and the price of each product changes in real time. A huge number of concurrent access requests for the product database results in long response time on the web page of each branch. In centralized caching mode, the bandwidth becomes a bottleneck and blocks the access requests for product prices.

Message Queue for Apache RocketMQ can reduce the page response time by means of large-scale cache synchronization for different branches. This satisfies customers' access requirements for product prices.

## 26.1.7. Limits

This topic describes the limits on specific metrics in Message Queue for Apache RocketMQ. To avoid system exception errors, do not exceed the limits when you use Message Queue for Apache RocketMQ.

The following table describes the limits.

Limits of Message Queue for Apache RocketMQ

| Item | Limit | Description |
| --- | --- | --- |
| The length of a topic name | 64 characters | A topic name can be up to 64 characters in length. If the name of a topic is more than 64 characters in length, messages in the topic cannot be subscribed to or sent. |

| Item | Limit | Description |
| --- | --- | --- |
| The size of a message | • A normal message or an ordered message: 4 MB<br>• A transactional message, a scheduled message, or a delayed message: 64 KB<br>Note Message attributes are part of a message. The size of message attributes cannot exceed 1 KB. | The size of a message cannot exceed the maximum size that is allowed for the message type. If the size of a message exceeds the corresponding limit, the message fails to be sent. |
| Message retention period | Three days | Messages can be retained for up to three days. After the three-day retention period ends, the system automatically deletes the messages. |
| Consumer offset resetting | Three days | Consumer offsets can be reset for messages that are retained within the previous three days. |
| The rate of transactions per second (TPS) for sending and receiving messages on a single Message Queue for Apache RocketMQ instance | For a cluster that consists of two primary brokers and two secondary brokers, a message of 1 KB size is used as a baseline. The following data is verified in the test environment:<br>• Messages that are sent and received per second: 30,000<br>Note A maximum of 10,000 messages can be sent per second. | N/A |
| Delay time of scheduled or delayed messages | 40 days | The msg.setStartDeliverTime parameter can be set to a point in time within the following 40 days. The unit of the parameter value is milliseconds. If the specified point in time is not within the 40 days, the message fails to be sent. |

# 26.1.8. Terms

This topic introduces the terms related to Message Queue for Apache RocketMQ to help you better understand and use this service.

***Topic***
A topic is used to classify messages. It is the primary classifier.

***Message***
In Message Queue for Apache RocketMQ, a message is a carrier that is used to transfer information.

***Message ID***
A message ID is a globally unique identifier for a message. Each message ID is automatically generated by the system.

***Message key***
A message key is a unique identifier for the service logic of a message. Each message key is set by a producer.

***Tag***
A tag is used to further classify messages within a topic. It is the secondary classifier.

***Producer***
A producer, also known as a message publisher, produces and sends messages.

***Producer instance***
An object instance of a producer. Different producer instances can run in different processes or on different machines. Producer instances can be shared among producer instance threads in the same process because these threads are secure.

***Consumer***
A consumer, also known as a message subscriber, receives and consumes messages.

***Consumer instance***
An object instance of a consumer. Different consumer instances can run in different processes or on different machines. A thread pool is configured for a consumer instance to consume messages.

***Group***
A group consists of producers or consumers that produce or consume messages of the same type and publish or subscribe to messages based on the same logic.

***Group ID***
A group ID identifies a group.

***Queue***
Each topic has one or more queues to store messages.

***Exactly-once delivery semantics***
Messages that are sent to the system can be consumed by consumers only once even when a message consumption retry occurs.

***Clustering consumption***
All consumers that are identified by the same consumer ID consume messages in an even manner. For example, if a topic contains nine messages and three consumer instances are identified by the same consumer ID, each of the three consumer instances consumes three messages in clustering consumption mode.

***Broadcasting consumption***

Each of the consumers that are identified by the same consumer ID consume each message once. For example, if a topic contains nine messages and three consumer instances are identified by the same consumer ID, each of the three consumer instances consumes the nine messages in broadcasting consumption mode.

***Scheduled message***

A producer sends a message to a RocketMQ broker and expects the message to be delivered to a consumer at a specified time in the future. The message is a scheduled message.

***Delayed message***

A producer sends a message to a RocketMQ broker and expects the message to be delivered to a consumer after a specified period of time. The message is a delayed message.

***Transactional message***

Message Queue for Apache RocketMQ provides a distributed transaction processing feature that is similar to X/Open XA to ensure transaction consistency by using transactional messages.

***Ordered message***

An ordered message is a message that is published and consumed in order. Ordered messages in Message Queue for Apache RocketMQ are classified into globally ordered messages and partitionally ordered messages.

***Ordered delivery of messages***

For a specified topic, a producer sends messages in specific order.

***Ordered consumption of messages***

For a specified topic, a consumer receives messages in first in, first out (FIFO) order.

***Globally ordered message***

All messages of a specified topic are published and consumed in strict FIFO order.

***Partitionally ordered message***

All messages of a specified topic are segmented by the sharding key. Messages in the same shard are published and consumed in strict FIFO order. A sharding key is a key field that is used in ordered messages to distinguish different shards. It is completely different from the key that is used in regular messages.

***Message accumulation***

A producer has sent messages to a RocketMQ broker but a consumer cannot consume all the messages in a short period of time due to limited consumption capability. Therefore, unconsumed messages are stored on the RocketMQ broker. This process is called message accumulation.

***Message filtering***

A subscriber can filter messages by tag so that the subscriber receives only the filtered messages. Messages are filtered on a RocketMQ broker.

***Message tracing***

The complete trace data of a message is aggregated by the time and location data of each node during the process from message delivery by a producer to message consumption by a

subscriber. Message tracing records the complete trace of a message from its delivery by a producer to a RocketMQ broker and then to a consumer. This facilitates troubleshooting.

***Consumer offset reset***

The timeline is used as coordinates. A subscriber can reset the consumption progress of a topic that is subscribed to over the time span of persistent message storage. By default, messages are stored for three days. After the consumer offset is reset, the subscriber can receive messages that are sent by a producer to a RocketMQ broker after the specified point in time.

# 27.Apsara Stack Security

# 27.1. Product Introduction

## 27.1.1. What is Apsara Stack Security

Apsara Stack Security is a solution that provides a full suite of security features, such as network, server, application, data, and security management to protect Apsara Stack assets.

### Background information

Traditional security solutions for IT services use hardware products such as firewalls and intrusion prevention systems (IPSs) to detect attacks on network perimeters and protect networks against attacks.

Cloud computing features low costs, on-demand flexible configuration, and high resource utilization. As cloud computing develops, an increasing number of enterprises and organizations use cloud computing services instead of traditional IT services. 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

| Security domain | Service name | Description |
| --- | --- | --- |
| Security management | Threat Detection Service | Monitors traffic and overall security status to audit and centrally manage assets. |
| | Cloud Security Scanner | Uses AI technologies to help enterprises identify security risks at the earliest opportunity. |
| | Security Operations Center (SOC) | Manages the overall security operations of the cloud environment. You can build a closed-loop security operations system that features risk prediction and discovery, defense control, detection and analysis, and response management. SOC is a cloud-native solution. |
| Server security | Server Guard | Protects Elastic Compute Service (ECS) instances against intrusions and malicious code. |
| | Server Security | Protects physical servers against intrusions. |
| | Container Protection | Protects containers and runtime environments against intrusions. |

| Security domain | Service name | Description |
| --- | --- | --- |
| 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. |
| | Network Detection and Response | Detects and responds to network attacks at the Internet border and internal network border based on a variety of threat detection engines and threat intelligence data. |
| Data security | Sensitive Data Discovery and Protection (SDDP) | Prevents data leaks and helps your business system meet compliance requirements. |
| O&M audit | Security Audit | Summarizes and analyzes logs so that security auditors can detect and eliminate risks in time. |
| 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. |

# 27.1.2. Benefits

Apsara Stack Security is a leading service in the cloud security industry and has received various certifications from relevant authorities. Apsara Stack Security fully protects the security of the Apsara Stack environment based on advanced security systems and security technologies.

## Leading service in the cloud security industry

The Apsara Stack Security team accumulated a large amount of security experience by protecting all internal business systems of Alibaba Group from 2005. After the service was released in 2011, Apsara Stack Security became a leading service that provides comprehensive protection to ensure cloud security.

## Mature systems and advanced technologies

Apsara Stack Security is a service that is developed based on ten years of protection experience. After the service protected the internal business systems of Alibaba Group for more than 10 years, Alibaba Group obtained a large number of security research achievements, security data, and security operations methods, and built a professional cloud security team. Apsara Stack Security is developed based on the rich experience of the security team to help enterprises build sophisticated systems that enhance 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 services

| Benefit | Traditional security service | Apsara Stack Security |
| --- | --- | --- |
| Comprehensive industry-leading security capabilities among Internet enterprises | A traditional security service provider offers only limited services and features, and cannot provide a comprehensive security protection system. | Alibaba Group accumulated a large number of intelligence sources based on multiple years of attack prevention experience. As a result, the service can detect common Internet attacks including zero-day exploits, and provide comprehensive security capabilities. |
| Early risk detection | Traditional security services cannot detect risks because they do not have experience in monitoring a wide array of services. | Apsara Stack Security can detect and quickly respond to critical vulnerabilities and security events to prevent security issues. |
| Big data modeling and analysis | Traditional security services cannot detect threats by scanning signatures. The traditional log analysis feature can be used to only collect data and report analysis results. You cannot use this feature to perform data modeling and analysis. | Apsara Stack Security implements big data modeling and analysis 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 way, Apsara Stack Security provides additional insights into the security of the system. |
| Scalability and decoupling from hardware | Traditional security services are developed based on existing hardware devices. These security services require virtual hosts that are created on virtualization platforms. | • Hardware and software decoupling: All modules are developed based on the cloud computing architecture and the common x86 architecture for hardware. As a result, the modules do not require specific hardware.<br>• Scalability: You can scale out hardware devices to increase performance without the need to change the network architecture. |
| Comprehensive protection system that supports unified detection and response | Traditional security services add devices to improve security capabilities. The devices can collect only device logs and status data, and display the data on the management platform. You cannot integrate the devices to use additional features. | Apsara Stack Security provides comprehensive Internet protection to ensure the security of networks, servers, applications, data, and user identities. The security modules automatically interact with each other to respond to security issues and share intelligence. |

| Benefit | Traditional security service | Apsara Stack Security |
| --- | --- | --- |
| Compatibility with all data center environments and decoupling from specific cloud platforms | Most traditional security services are provided as hardware appliances. Due to this reason, the services are incompatible with the cloud platforms that adopt Software Defined Network (SDN) technology. | Based on the interactions between servers and the operating system, Apsara Stack Security performs data analysis to detect threats at the network perimeter. This helps ensure that Apsara Stack Security is compatible with all data center environments and prevents the complex network topology in the data centers. |

# 27.1.3. Service 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 based on machine learning and data modeling. This module also provides up-to-date information about ongoing attacks to help you monitor the security status of your business.

- Network Detection and Response

  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 based on traffic mirroring. The analysis results are used by other Apsara Stack Security modules.

- Cloud Security Scanner

  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 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 provides security features to protect Elastic Compute Service (ECS) instances. The features include 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.

- Security Audit

This module collects database logs, server logs, and operation logs of the user console and IT administrator console, and network device logs in Apsara Stack. This module stores and analyzes these logs, and triggers alerts for suspicious events.

- Web Application Firewall (WAF)

  This module protects web applications against common web attacks reported by Open Web Application Security Project (OWASP), such as SQL injections, cross-site scripting (XSS), exploits of vulnerabilities in web server plug-ins, trojan uploads, and unauthorized access. This module also blocks a large number of malicious requests to prevent 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, management of access control policies, 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 DDoS attacks.

- Sensitive Data Discovery and Protection (SDDP)

  This module uses the big data analytics capabilities and 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 dataflows, and detect abnormal activities. This module provides visible, controllable, and industry-compliant security protection for your sensitive data by using precise detection and analysis.

- Container Protection

  This module helps monitor all exceptions that occur in containers and their infrastructure. This module provides capabilities such as intelligent learning of protection policies, exception detection containers and their infrastructure, automatic and custom exception handling, and threat source tracing. This module helps prevent intrusions by exploiting zero-day vulnerabilities or monitoring blind spots.

# 27.1.4. Features

## 27.1.4.1. 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.<br>• 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.<br>• 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.<br>• Helps you customize and optimize security policies. |
| 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. |

# 27.1.4.2. Apsara Stack Security Standard Edition

# 27.1.4.2.1. Overview

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.

Apsara Stack Security Standard Edition provides the following modules: Network Detection and Response, Cloud Security Scanner, Server Guard, Security Audit, Web Application Firewall (WAF), Threat Detection Service (TDS), and Security Operations Center (SOC).

# 27.1.4.2.2. Network Detection and Response

This topic describes the features of Network Detection and Response.

| Feature | Description |
| --- | --- |
| 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. |

# 27.1.4.2.3. Cloud Security Scanner

This topic describes the features of Cloud Security Scanner.

| Feature | Description |
| --- | --- |
| 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, display the details about the availability of an asset, and display basic monitoring information about a monitored website based on custom alert rules. |

| Feature | Description |
| --- | --- |
| 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. |
| 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 behavior and key enterprise information. |

# 27.1.4.2.4. Server Guard

**This topic describes the features of Server Guard.**

| Feature | Description |
| --- | --- |
| 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.<br>• 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. |

# 27.1.4.2.5. Security Audit

**This topic describes the features of Security Audit.**

| Feature | Description |
| --- | --- |
| Raw log collection | Collects the following types of logs:<br>• Database and server logs<br>• Operation logs of the user console and the IT administrator console<br>• Network device logs |
| Audit log query | Allows you to query audit logs by audit type, audit object, operation type, operation risk level, alert, or creation time. Full-text search is supported. |
| Policy configuration | Allows you to configure audit rules based on the following parameters: initiator, object, command, result, and cause. This feature identifies risky operations in raw logs and generates alerts. |

# 27.1.4.2.6. Web Application Firewall

**This topic describes the features of Web Application Firewall (WAF).**

| Feature | Description |
| --- | --- |
| Protection overview | Provides the following capabilities:<br>• Protection overview: provides statistics on protection for the last 24 hours and the last 30 days.<br>• Access status monitor: displays the top 100 access requests in real time.<br>• Export protection report: allows you to export daily reports, weekly reports, and reports of scheduled tasks.<br>• Attack detection statistics: provides statistics on attack detection. |
| Protection logs | Provides the following capabilities:<br>• 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.<br>• HTTP flood protection 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.<br>• System operation logs: provides system operation logs, including usernames, operations, and IP addresses.<br>• Access logs: provides access logs, including the access addresses, destination IP addresses, source IP addresses, request methods, and HTTP status codes of requests. |

| Feature | Description |
| --- | --- |
| Protection configuration | Provides the following capabilities:<br>• Protection site management: allows you to create, delete, modify, enable, and disable feature forwarding proxies of a protected site.<br>• Custom rules: allows you to create, delete, enable, and disable custom rules. This implements fine-grained HTTP access control for websites.<br>• Website protection policies:<br>  ○ Supports decoding methods, such as URL decoding, JSON parsing, Base64 decoding, hexadecimal conversion, backslash unescape, XML parsing, PHP deserialization, and UTF-7 decoding.<br>  ○ 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.<br>  ○ 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.<br>  ○ Allows you to enable HTTP response detection and configure the length of the response body in detection rules.<br>  ○ Allows you to configure the length of the request body in detection rules.<br>  ○ Allows you to enable or disable detection timeout settings.<br>• 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. |

# 27.1.4.2.7. Threat Detection Service

**This topic describes the features of Threat Detection Service (TDS).**

| Feature | Description |
| --- | --- |
| 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. |

| Feature | Description |
| --- | --- |
| 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 manual checks and periodic checks that automatically run. 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.<br>• 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. |

# 27.1.4.2.8. Security Operations Center

**This topic describes the features of Security Operations Center (SOC).**

| Feature | Description |
| --- | --- |
| Operations center | Provides scenario-based operation capabilities such as regular security inspection, security monitoring during major events and cybersecurity drills, and real-time attack and defense. |
| Situation monitoring | Displays information about the global security posture of Apsara Stack and all cloud tenants. The information includes network attack statistics, abnormal behavior statistics, asset vulnerability statistics, security risk trends of Apsara Stack and tenants, distribution of abnormal behavior by type, most recent alerts that are generated for abnormal behavior, distribution of network attacks by type, most recent alerts on network attacks, and distribution of assets by type. This module is used to monitor the network security of Apsara Stack. |
| Risk analysis | Allows you to trace sources of risks and identify, determine, and obtain evidence of risks that occur on Apsara Stack. The feature also provides capabilities such as threat event determination based on Indicators of Compromise (IOC), vulnerability analysis, network traffic analysis, and threat intelligence investigation. |

| Feature | Description |
| --- | --- |
| Asset management | Allows you to manage risks that occur on cloud-based hosts and virtualized assets. The feature helps you obtain the details of both assets on Apsara Stack and assets of tenants. The feature also helps you obtain the details of the risks that occur on assets, such as the risks caused by network attacks, abnormal behavior, and vulnerabilities. |
| SOAR | Used as a key module and can be used as an independent service. Security Orchestration Automation Response (SOAR) allows you to orchestrate automated security operations, security analysis, and risk handling. SOAR enables you to handle risks that occur on Apsara Stack based on specific playbooks. This helps improve threat response speed, reduces the threat response time, and avoids excessive workloads that are caused by repetitive security operations. |
| Log management | Allows you to manage all raw logs that are collected to SOC. You can view statistics of full logs, perform associated log retrieval and queries on logs, audit logs, connect log sources to SOC, and deliver logs. |
| Report management | Allows security operations administrators to export reports, creates reports on a daily basis, weekly basis, and monthly basis, automatically exports the reports, and automatically sends the reports to the specified email addresses. |
| Rule management | Allows you to manage rules to detect and block risks. The feature also allows you to configure the rules to perform operations such as security analysis and risk handling. |
| Operations | Allows you to configure basic settings of SOC, including security audit, storage management, and the IP address library. |

# 27.1.4.3. Optional security services

# 27.1.4.3.1. Overview

**Apsara Stack Security provides a variety of optional security services. Each service provides multiple features.**

**The optional services include Anti-DDoS Service and Sensitive Data Discovery and Protection (SDDP).**

# 27.1.4.3.2. Anti-DDoS Service

**This topic describes the features of Anti-DDoS Service.**

| Feature | Description |
| --- | --- |
| Traffic scrubbing against DDoS attacks | Detects and defends against attacks, such as SYN flood, ACK flood, ICMP flood, UDP flood, NTP flood, DNS flood, and HTTP flood attacks. |
| DDoS attack display | Allows you to search for DDoS attack events by IP address, status, and event information. |

| Feature | Description |
| --- | --- |
| DDoS traffic analysis | Allows you to monitor and analyze the traffic of a DDoS attack. You can also view the protocol of attack traffic and the top 10 IP addresses from which most attacks are launched. |

# 27.1.4.3.3. Sensitive Data Discovery and Protection

This topic describes the features of Sensitive Data Discovery and Protection (SDDP).

| Feature | Description |
| --- | --- |
| 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 | Allows you to create audit rules to intelligently perform audits on services such as OSS, MaxCompute, and ApsaraDB RDS. |
| Data permission management | Displays departments and users in a hierarchical structure, 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. |
| Access authorization | Supports department-specific authorization and protects the data assets of authorized departments. |

# 27.1.4.3.4. Container Protection

This topic describes the features of Container Protection.

| Feature | Description |
| --- | --- |
| Management of container assets | Provides various security management solutions for container assets from the cluster perspective. This helps meet the requirements of different users. |
| Image management | Provides the asset management module that supports image-based risk tracing and protection capabilities. |

| Feature | Description |
| --- | --- |
| Container image scan | Detects and identifies high-risk system vulnerabilities, application vulnerabilities, malicious samples, configuration risks, and sensitive data in images. |
| Intrusion alerting | Generates different types of alerts for assets in real time. The types of alerts include the alerts for web tampering, suspicious processes, webshells, unusual logons, and malicious processes. |
| Log retrieval | Allows you to query logs for logons, brute-force attacks, snapshots of processes, network connections, snapshots of listening ports, snapshots of accounts, and process startup. You can use prefixes to query logs in fuzzy match mode. |

# 27.1.5. Restrictions

None

# 27.1.6. 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.

# 28.MaxCompute

# 28.1. Product Introduction

## 28.1.1. 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.
- Stores audit logs and automatically dumps them to a specific server directory for long-term storage and management.

## 28.1.2. 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 |

| Service | Supported character set |
| --- | --- |
| PAI | UTF-8 |
| OSS | UTF-8 |
| 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. |

# 28.1.3. 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.

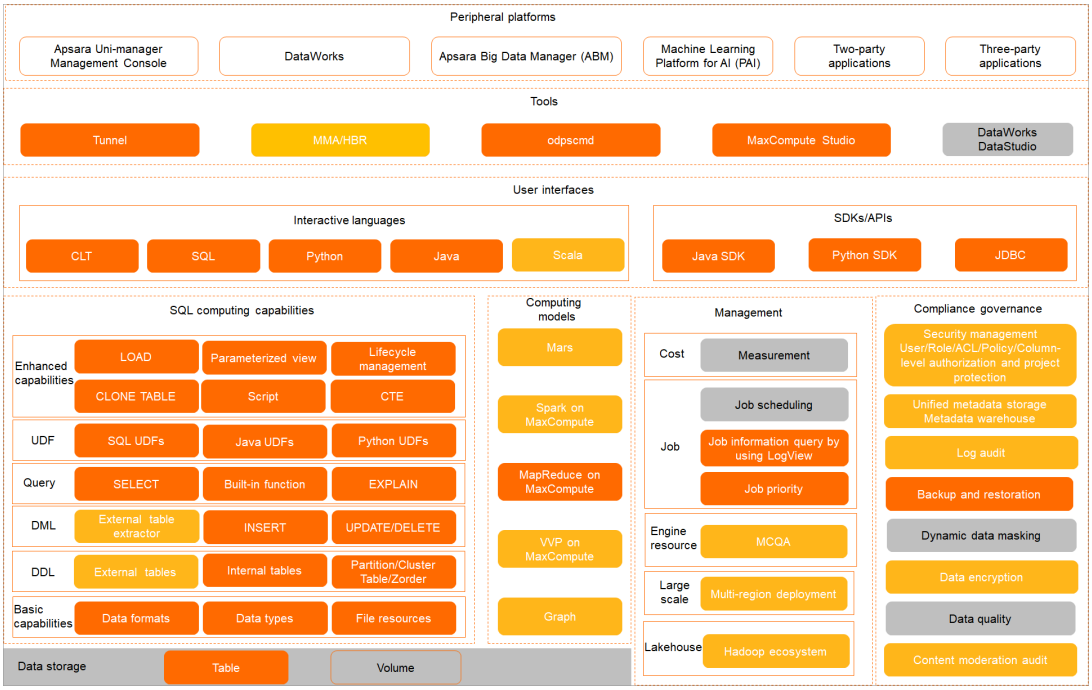
### Multi-device support

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.

## 28.1.4. 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 |
| --- | --- |
| Peripheral platforms | MaxCompute supports the following peripheral platforms:<br>• Apsara Uni-manager Management Console: a unified and intelligent O&M platform. For more information, see *Apsara Uni-manager Management Console User Guide*.<br>• 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 *DataWorks Technical White Paper*.<br>• Apsara Big Data Manager (ABM): provides an easy method for field engineers to manage MaxCompute. For more information, see *Apsara Big Data Manager Technical White Paper*.<br>• Machine Learning Platform for AI (PAI): a machine learning algorithm platform based on MaxCompute. For more information, see *Machine Learning Platform for AI Technical White Paper*.<br>• Two-party applications: other Alibaba Cloud services supported by MaxCompute, such as DataV.<br>• Three-party applications: other services that are compatible with MaxCompute. |
| Tools | MaxCompute supports the following tools:<br>• 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 *Tunnel* in *MaxCompute Product Introduction*.<br>• 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.<br>• Hybrid backup recovery (HBR): integrates data backup and migration capabilities of Apsara Stack.<br>• odpscmd: the MaxCompute client. For more information, see *Client* in *MaxCompute User Guide*.<br>• 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.<br>• DataWorks DataStudio: a visualized development platform provided by DataWorks. For more information, see *DataWorks User Guide*. |

| Category | Description |
| --- | --- |
| User interfaces | MaxCompute supports the following interfaces:<br>• Interactive languages: CLI, SQL, Python, Java, and Scala.<br>• SDKs and APIs: SDK for Java, SDK for Python, and Java Database Connectivity (JDBC).<br>For more information, see *MaxCompute Developer Guide*. |
| SQL computing capabilities | MaxCompute supports the following SQL computing capabilities:<br>• Enhanced capabilities: support LOAD, parameterized view, lifecycle management, and CLONE TABLE.<br>• User-defined functions (UDFs): include SQL UDFs, Java UDFs, and Python UDFs.<br>• Query: the query operations, such as SELECT and EXPLAIN statements and built-in functions.<br>• Data manipulation language (DML) statements: include INSERT, UPDATE, and DELETE.<br>• DDL statements: allow you to create internal tables, external tables, clustered tables, and partitioned tables.<br>• Basic capabilities: support multiple data types and data formats and allow you to upload resource files.<br>For more information, see *MaxCompute SQL* in *MaxCompute User Guide*. |
| Computing models | MaxCompute supports the following computing models:<br>• 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 *Mars* in *MaxCompute User Guide*.<br>• 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 *Spark on MaxCompute* in *MaxCompute User Guide*.<br>• MapReduce on MaxCompute: allows you to run MapReduce jobs on MaxCompute. For more information, see *MaxCompute MapReduce* in *MaxCompute User Guide*.<br>• 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 *VVP On MaxCompute* in *MaxCompute User Guide*.<br>• Graph: a processing framework designed for iterative graph computing. For more information, see *MaxCompute Graph* in *MaxCompute User Guide*. |

| Category | Description |
| --- | --- |
| Management | MaxCompute can be managed from the following aspects:<br>• Cost: measures resource usage.<br>• 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.<br>• Engine resource: supports high-performance MaxCompute Query Acceleration (MCQA).<br>• Large scale: allows you to deploy MaxCompute clusters across regions.<br>• 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.<br>For more information, see *MaxCompute Operations and Maintenance Guide*. |
| Compliance governance | MaxCompute allows you to use the following methods for compliance governance:<br>• 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.<br>• Unified metadata storage: stores metadata in a centralized manner.<br>• Log audit: audits different log data of different users.<br>• Backup and restoration: allows you to back up and restore data from a storage system.<br>• Dynamic data masking: allows you to query data masking rules in DataWorks.<br>• Data quality: DataWorks provides an end-to-end platform that supports quality verification, notification, and management services for various heterogeneous data sources.<br>• Content moderation audit: uses the content moderation engine to identify and audit pornographic, violent, and illegal content.<br>For more information about security, see *MaxCompute Security White Paper*. |
| Data storage | MaxCompute stores data as tables or volumes. |

# 28.1.5. Features

## 28.1.5.1. Tunnel

### 28.1.5.1.1. 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.

# 28.1.5.1.2. 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

# 28.1.5.1.3. 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;
```

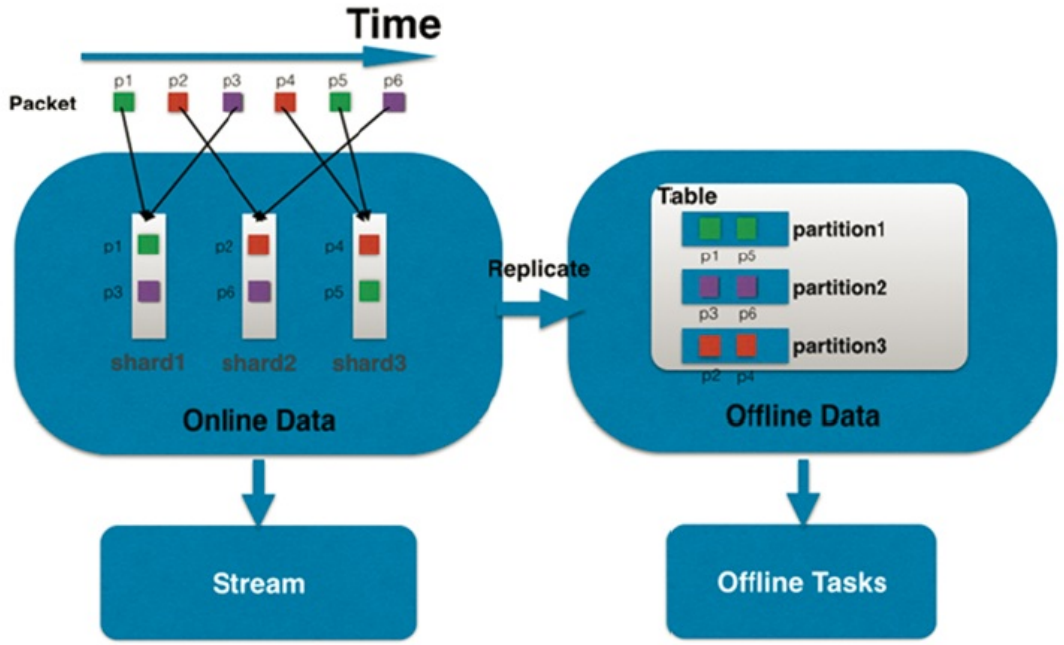
## Notes

- 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




# 28.1.5.2. SQL

## 28.1.5.2.1. 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.

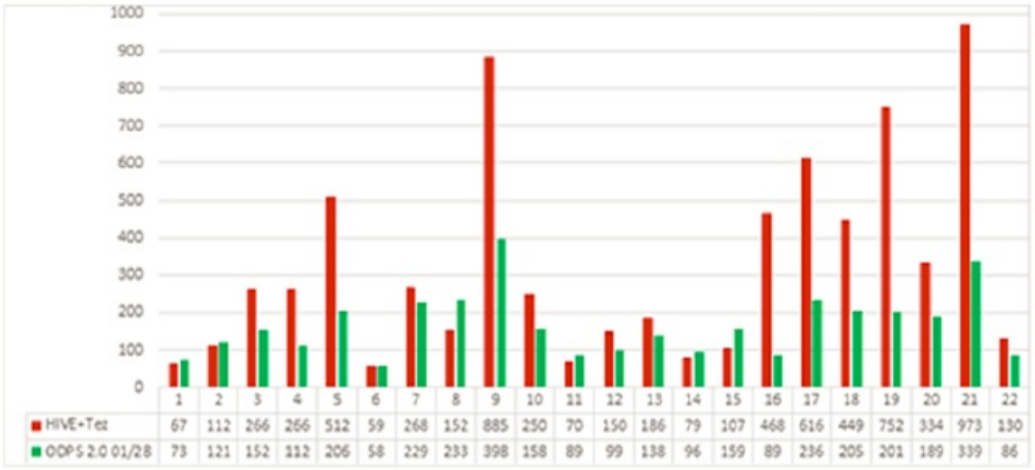
## 28.1.5.2.2. SQL characteristics

- 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.

## 28.1.5.2.3. Comparison with open source products

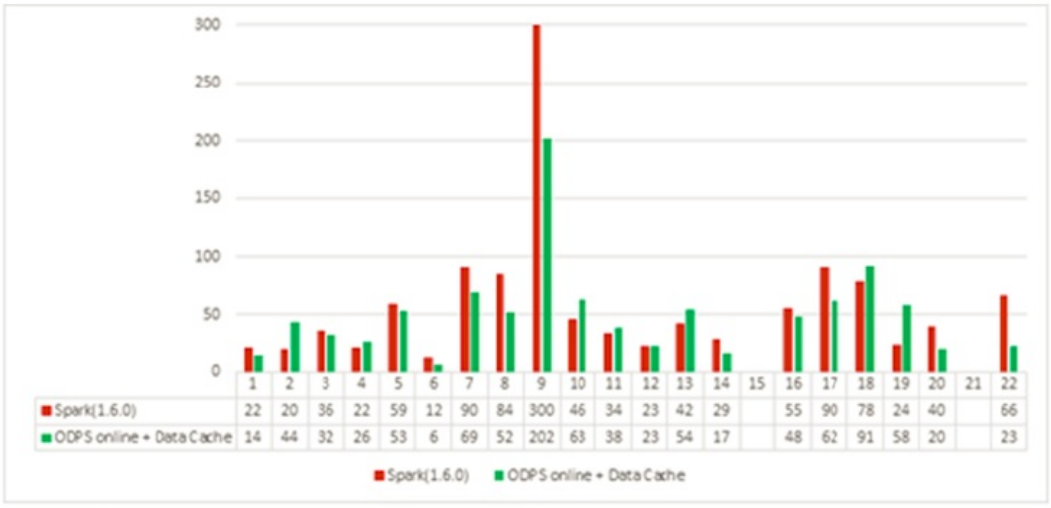
- TPC-H 1 TB data benchmark: Compared with Hive (Apache Hive-1.2.1-bin + Tez-UI-0.7.0 with CBO), MaxCompute has a 95.6% improvement in performance.

  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




# 28.1.5.3. MapReduce

## 28.1.5.3.1. 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.

## 28.1.5.3.2. MapReduce characteristics

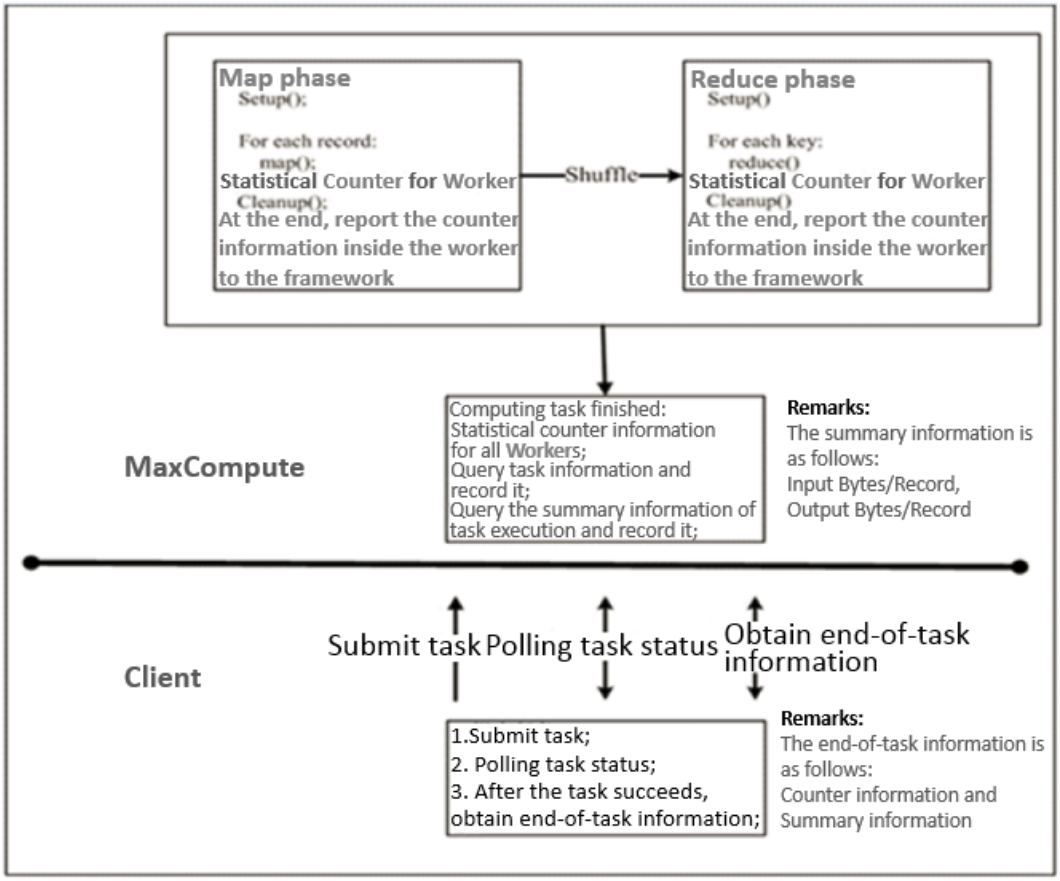
- It only supports the input and output of MaxCompute built-in data types.

- 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.

## 28.1.5.3.3. MaxCompute MapReduce process

The following figure shows the MapReduce process in MaxCompute:

MapReduce process




## 28.1.5.3.4. 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 (lRecord 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);
  }
```

# 28.1.5.4. Graph

## 28.1.5.4.1. 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.

## 28.1.5.4.2. 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.

## 28.1.5.4.3. 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.

## 28.1.5.5. 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

# 28.1.5.6. 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.

# 28.1.5.7. Enhanced features

# 28.1.5.7.1. Spark on MaxCompute

# 28.1.5.7.1.1. 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.

# 28.1.5.7.1.2. 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.

## 28.1.5.7.1.3. Spark features

The following table describes **Spark on MaxCompute** features.

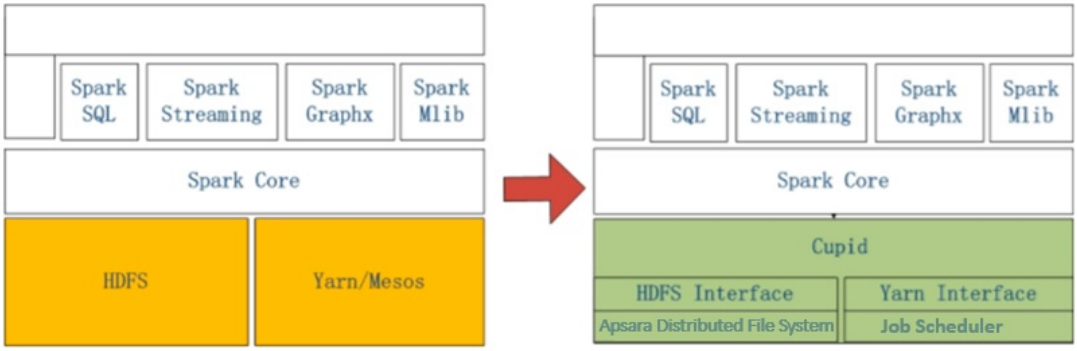
Features

| Type | Feature | Description |
| --- | --- | --- |
| Distributed cluster | Cluster deployment<br>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<br>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. |

## 28.1.5.7.1.4. 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.

# 28.1.5.7.1.5. 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.

# 28.1.5.7.2. Elasticsearch on MaxCompute

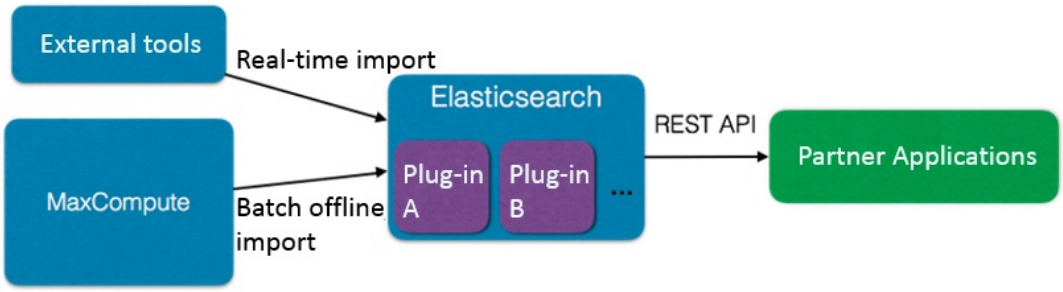
# 28.1.5.7.2.1. Overview

**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




# 28.1.5.7.2.2. 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.

## 28.1.5.7.2.3. Elasticsearch features

**Elasticsearch on MaxCompute** features are described as follows:

Features

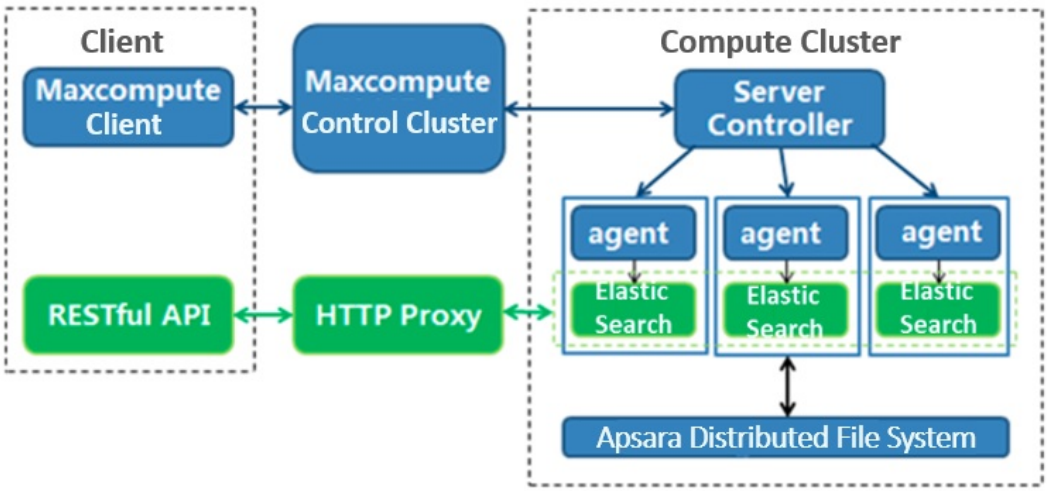
| Type | Feature | Description |
| --- | --- | --- |
| Distributed cluster | Cluster deployment<br>Cluster monitoring | Provide an O&M platform to monitor clusters, nodes, and indexes. |

| Type | Feature | Description |
| --- | --- | --- |
| Retrieval management | Index configuration management<br>Structure definition and index rebuilding | Provide a retrieval management platform and support configuration. |
| Full-text retrieval | Retrieval<br>Sorting<br>Statistical analysis | The features are provided through RESTful APIs. |
| Data collection | Elasticsearch data import APIs<br>MaxCompute data import tools<br>Full and incremental collection | Support a variety of interfaces to collect native data.<br>Provide integrated tools to import MaxCompute data. |
| Service authentication | Service-level user authentication | Allow you to configure user authentication in a centralized manner. |

## 28.1.5.7.2.4. 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.

# 28.1.5.7.2.5. 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.

# 28.1.5.8. Mlulti-region deployment

# 28.1.5.8.1. 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.

# 28.1.6. Scenarios

## 28.1.6.1. 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.

## 28.1.6.2. 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.

## 28.1.6.3. 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.

## 28.1.6.4. 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.

# 28.1.7. Limits

None.

# 28.1.8. 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.

# 28.1.9. Storage performance

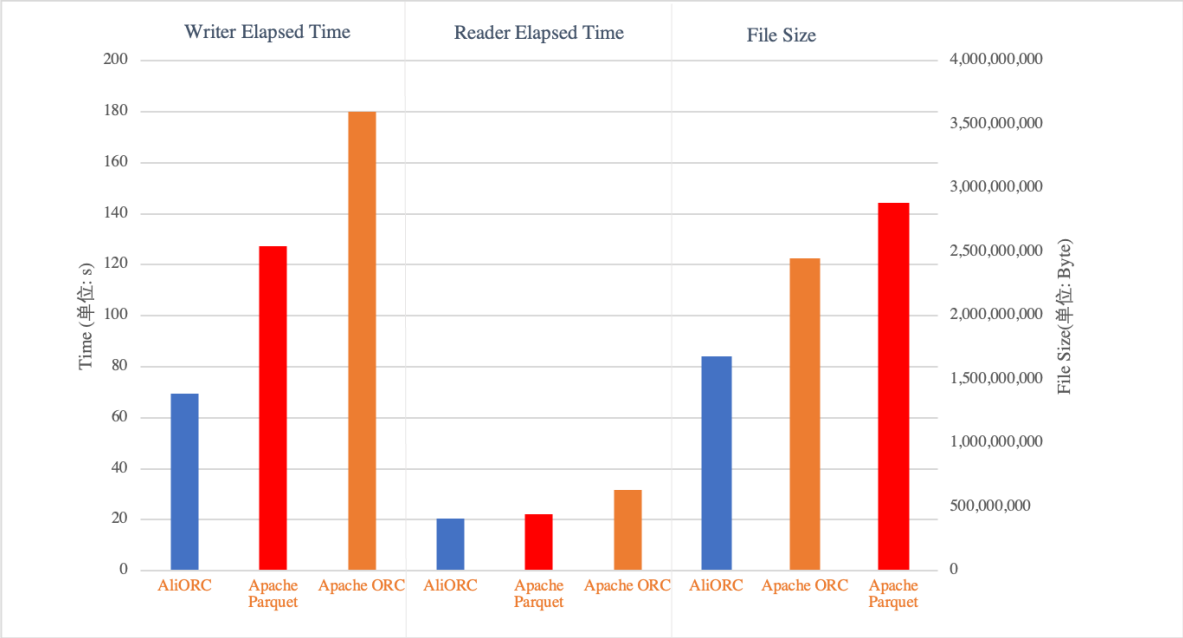
The data storage format of MaxCompute has been updated to AliORC since February 2020. This topic compares AliORC with Apache Optimized Row Columnar (ORC) and Apache Parquet and their compression ratios based on TPC Benchmark DS (TPC-DS) tests. The comparison helps you better understand the data storage 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.



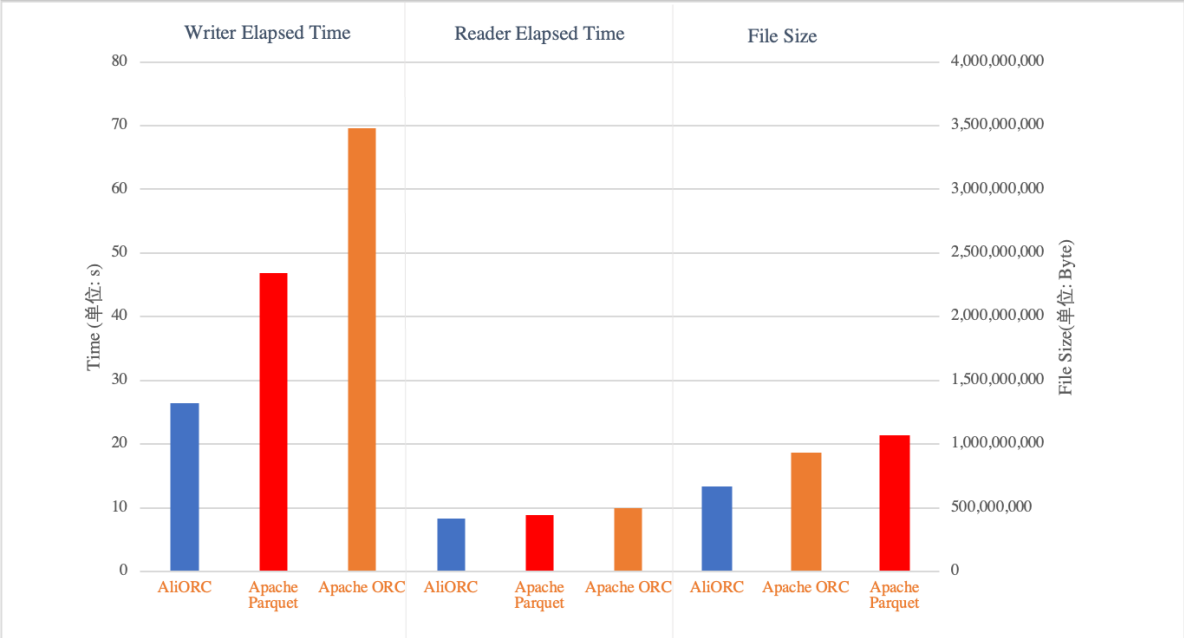

Parameters:

- 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 tpcd rows (official) | 10TB odps Table size | 1TB tpcds 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_demographi | 1920800 | 68663 | 1920800 | 68663 |
| date_dim | 73049 | 156661 | 73049 | 156661 |
| household_demograph | 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
```

# 29.Realtime Compute

## 29.1. Product Introduction

### 29.1.1. 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.

# 29.1.2. 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.

# 29.1.3. Differences between real-time computing and batch computing

## 29.1.3.1. 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.

## 29.1.3.2. 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.

## 29.1.3.3. 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.

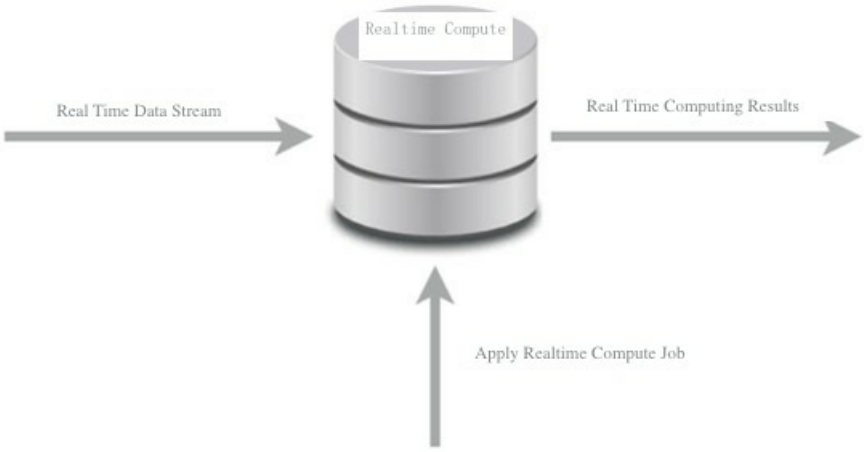
   The streaming data is continuously written to data stores in real time. You do not need to preload data for processing. Realtime Compute does not store real-time data that is continuously processed. The real-time data is discarded instantly after it has been processed.

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.

# 29.1.3.4. 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. |

| Item | Batch computing | Real-time computing |
| --- | --- | --- |
| 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.

# 29.1.4. 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.

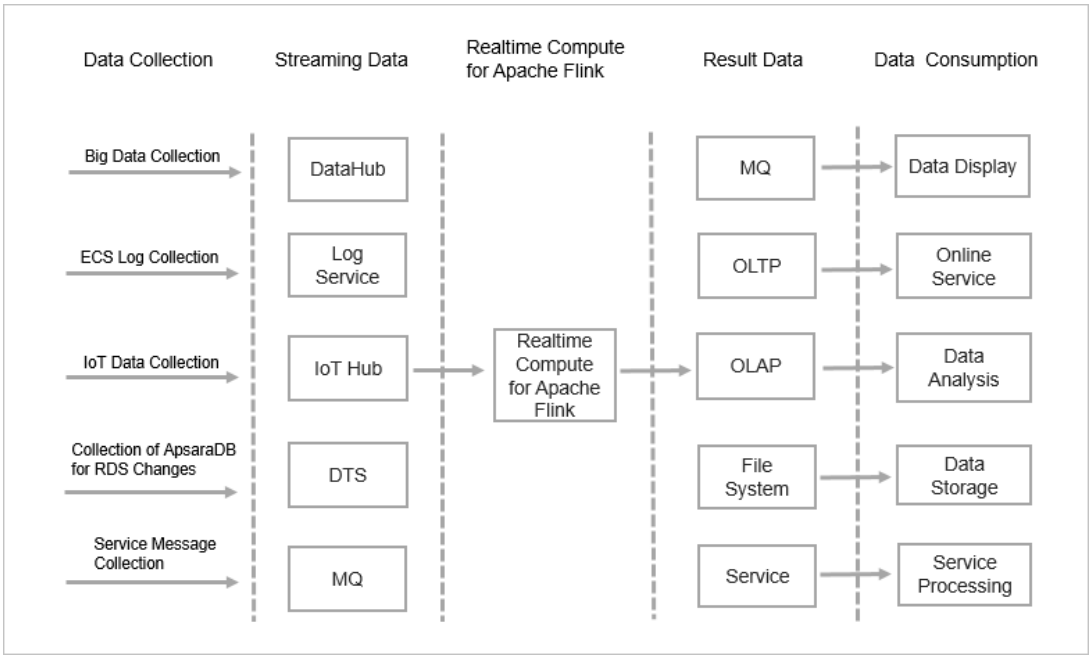
Supports CPUs, hard disks, memory, and NICs of different specifications in a single-component cluster without affecting cluster running performance. This guarantees maximum compatibility with existing devices.

# 29.1.5. Product architecture

## 29.1.5.1. 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

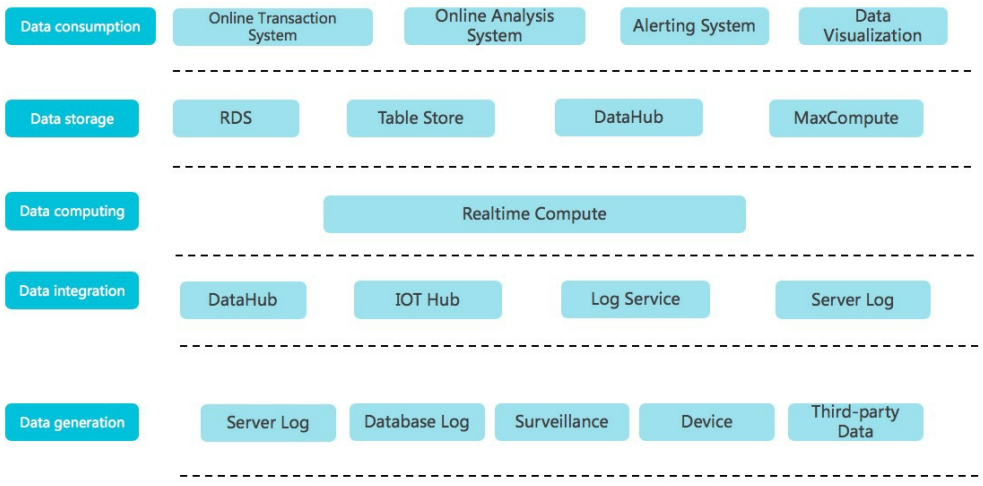
  After the result data of real-time computing is written to the sinks, you can consume the data by using custom applications.

  - 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.

# 29.1.5.2. 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

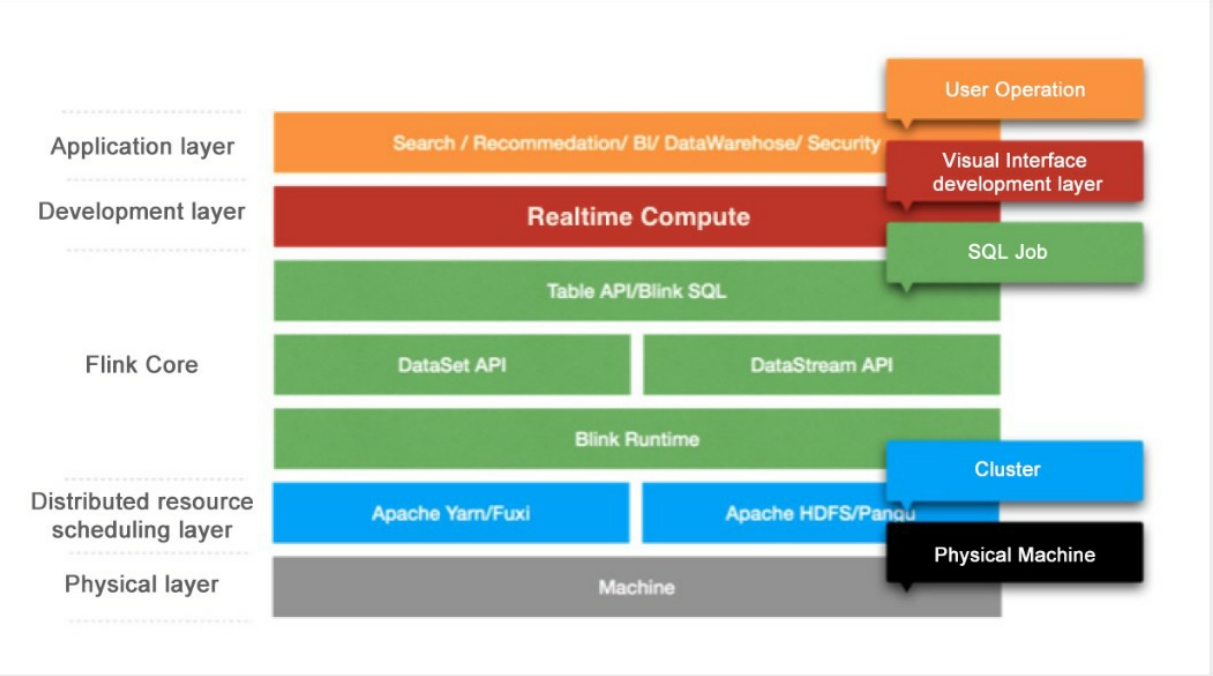
  Realtime Compute does not provide built-in data stores. Instead, it writes computing results to external data stores, such as relational databases, NoSQL databases, and online analytical processing (OLAP) systems.

- 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.

# 29.1.5.3. 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.

# 29.1.6. 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

# 29.1.7. 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.

# 29.1.8. Scenarios

## 29.1.8.1. 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.

## 29.1.8.2. 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.

# 29.1.8.3. 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 DOUBL
E) 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 DOUBL
E) 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;
```

# 29.1.8.4. 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 "seckilling" activities occur, such as 00:00 on November 11, hundreds of thousands of sales orders need to be processed in real time. The word "seckilling" 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.

## 29.1.8.5. 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.

# 29.1.9. Restrictions

None

# 29.1.10. 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.

# 30.DataHub
# 30.1. Product Introduction

## 30.1.1. 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).

## 30.1.2. 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.

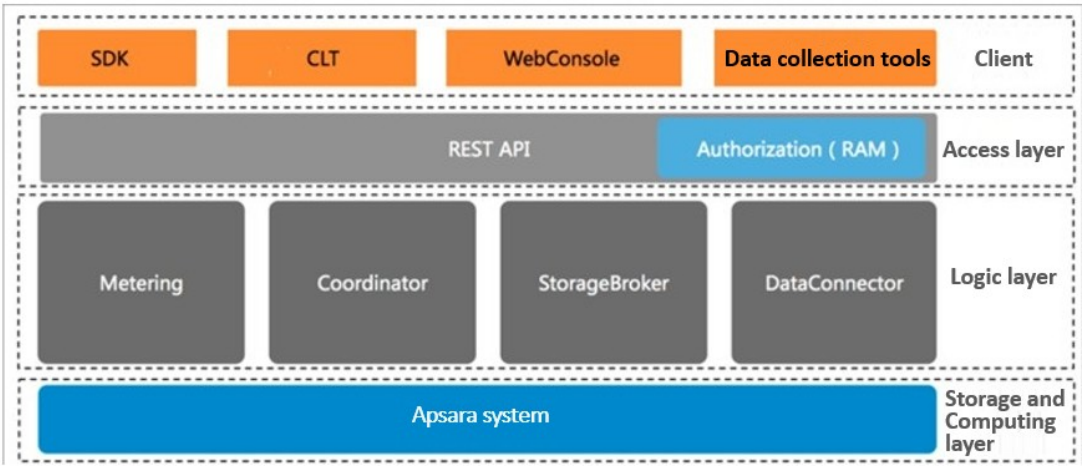
## Data security

- 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.

# 30.1.3. 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.

# 30.1.4. Features

## 30.1.4.1. 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.

## 30.1.4.2. 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.

## 30.1.4.3. 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.<br>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. |

## 30.1.4.4. 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.

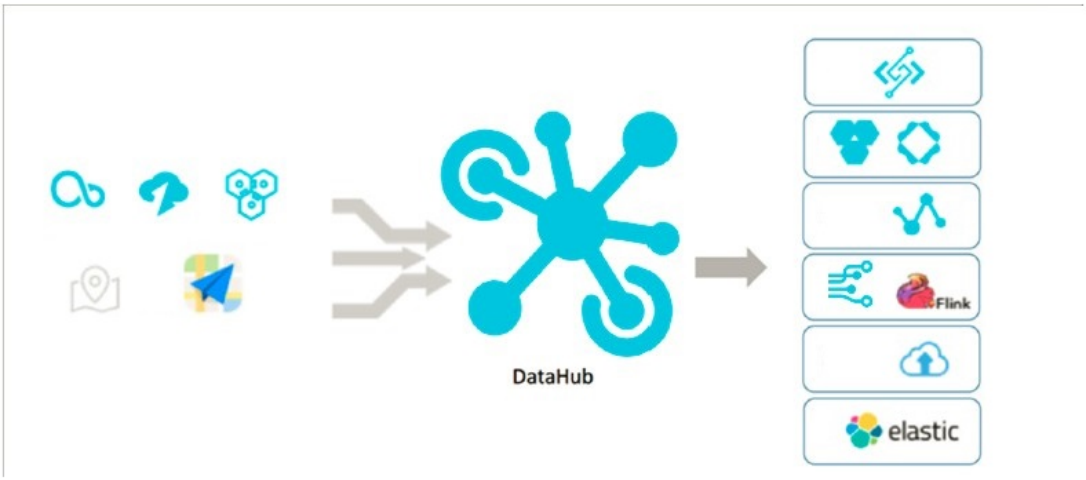
# 30.1.5. Scenarios

## 30.1.5.1. Overview

As a streaming data processing platform, DataHub can be used with various Alibaba Cloud products to provide one-stop data processing services.

# 30.1.5.2. Data uploading

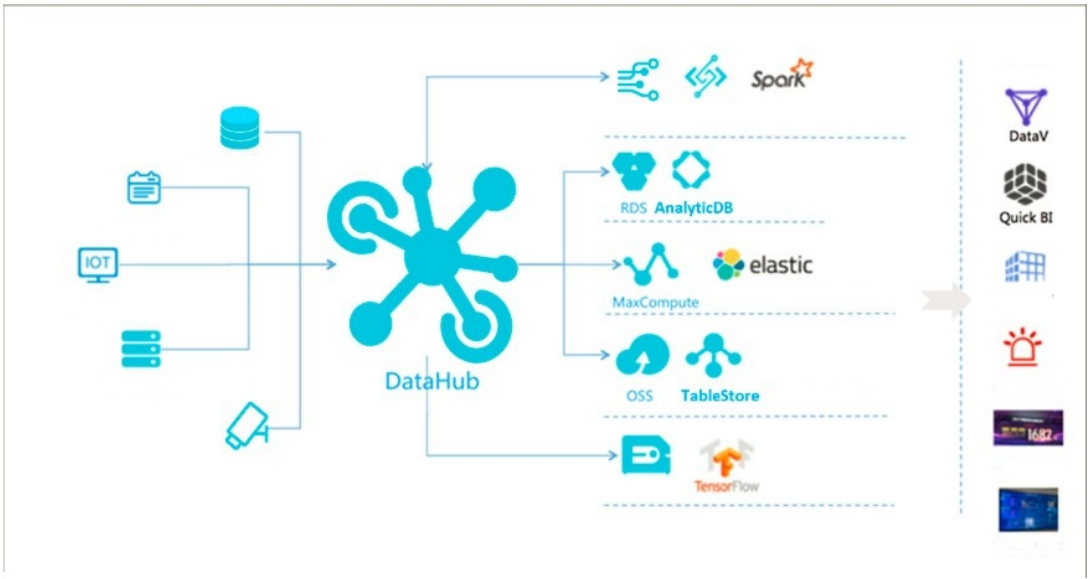
Data uploading




DataHub is connected to other Alibaba Cloud services, saving you the trouble of uploading the same data to different platforms.

# 30.1.5.3. Data collection

Data collection

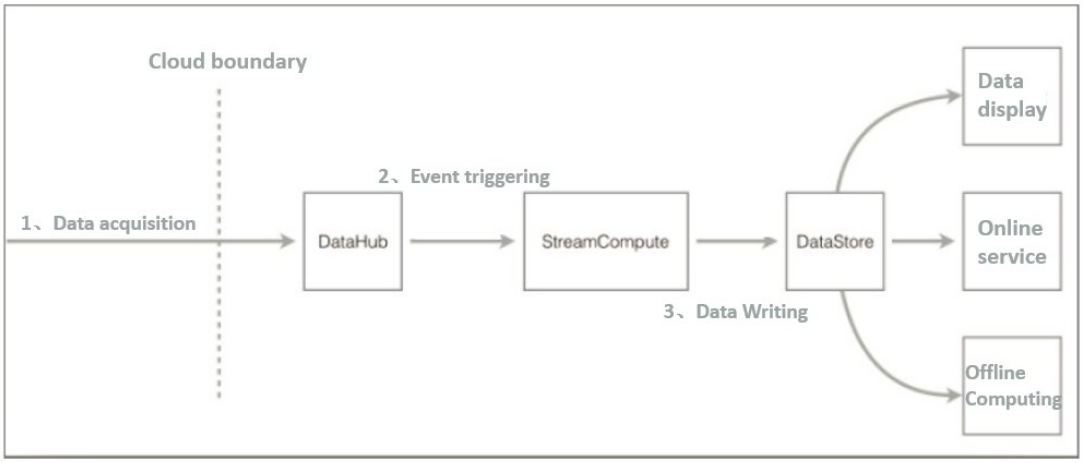



DataHub provides several types of data collection tools for you to write your data into DataHub. DataHub supports log collection from Logstash and Fluentd, and binary log collection from Data Transmission Service (DTS) and Oracle GoldenGate (OGG). DataHub also supports the collection of surveillance videos through GB28181.

# 30.1.5.4. 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




## 30.1.5.5. 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.

## 30.1.5.6. Data archiving

You can create a DataConnector to periodically archive data in DataHub to MaxCompute.

# 30.1.6. 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. |

| Item | Limit | Description |
| --- | --- | --- |
| 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. |
| 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. |

# 30.1.7. 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.<br>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.

# 31.DataWorks

# 31.1. Product Introduction

## 31.1.1. 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.

## 31.1.2. 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.

# 31.1.3. 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.

## 31.1.3.1. 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.

## 31.1.3.2. 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.

## 31.1.3.3. 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.

# 31.1.3.4. 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.

# 31.1.4. Services

# 31.1.4.1. 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.

# 31.1.4.2. DataStudio

# 31.1.4.2.1. 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.

# 31.1.4.2.2. 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.

# 31.1.4.2.3. Solutions

In a DataWorks workspace, you can group multiple workflows in a solution.

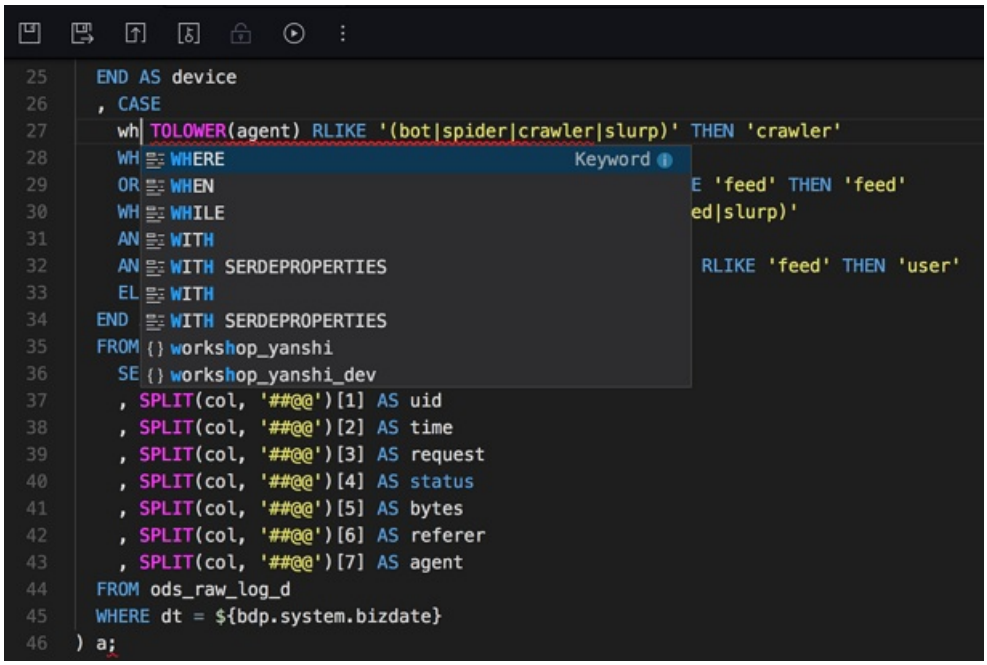
You can add one or more workflows to one solution so that you can manage them as a whole. In addition, a workflow can be added to multiple solutions, allowing you to assess your business based on solutions.

# 31.1.4.2.4. 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.




## 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.

# 31.1.4.2.5. 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.

# 31.1.4.3. Administration

# 31.1.4.3.1. 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.

# 31.1.4.3.2. 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.

# 31.1.4.3.3. 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.

# 31.1.4.3.4. 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.

# 31.1.4.3.5. 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.

# 31.1.4.4. 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.

## 31.1.4.5. 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.

## 31.1.4.6. 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.

# 31.1.4.7. 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.

# 31.1.4.8. 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.

## 31.1.4.9. 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. |

# 31.1.4.10. 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.

# 31.1.4.11. Data Protection

## 31.1.4.11.1. 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.

## 31.1.4.11.2. 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.

## 31.1.4.11.3. 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.

## 31.1.4.11.4. 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.

## 31.1.4.11.5. 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.

## 31.1.4.11.6. 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.

## 31.1.4.11.7. 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.

## 31.1.4.11.8. 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.

## 31.1.4.11.9. 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.

## 31.1.4.11.10. 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.

## 31.1.4.11.11. 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.

# 31.1.5. Scenarios

## 31.1.5.1. 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.

## 31.1.5.2. 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.

## 31.1.5.3. 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.

# 31.1.6. Limits

None.

# 31.1.7. 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.

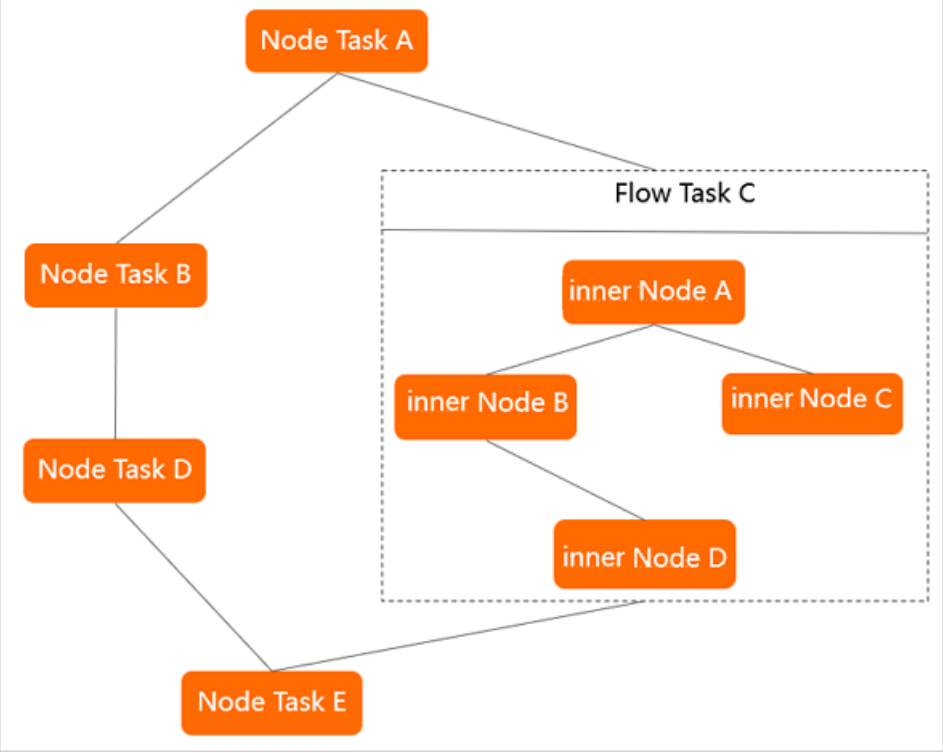
## node

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). |

| Node type | Description |
| --- | --- |
| 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.<br>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.<br>**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.

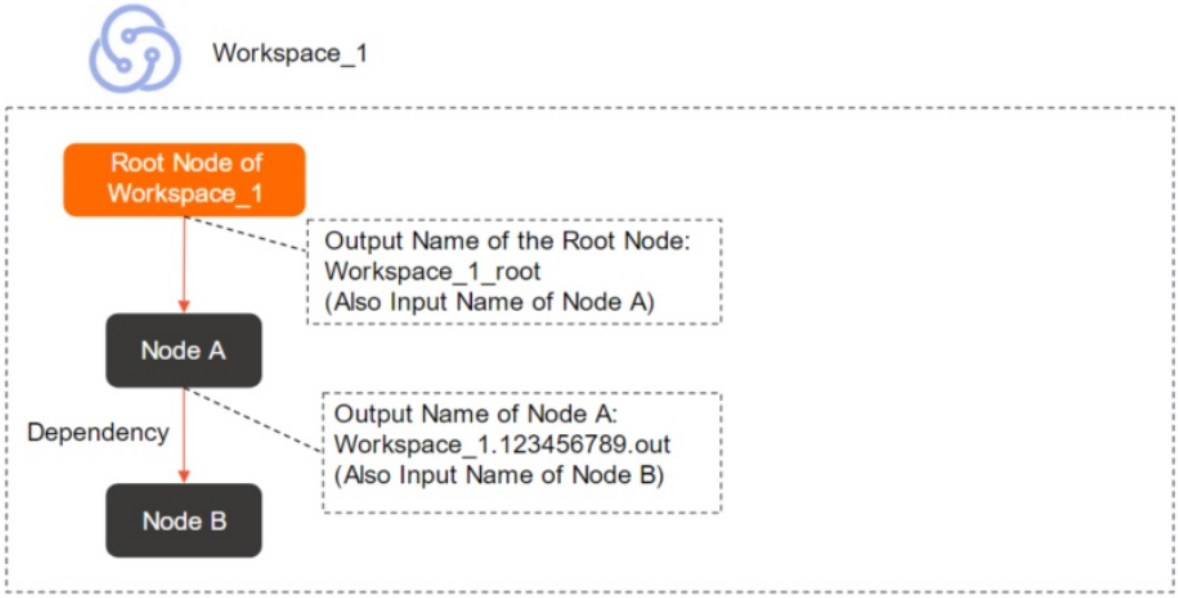
## resource and function

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.

# 32.Apsara Big Data Manager (ABM)

## 32.1. Product Introduction

### 32.1.1. 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
- Realtime Compute
- Quick BI
- DataHub

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.

### 32.1.2. 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.

### 32.1.3. Architecture

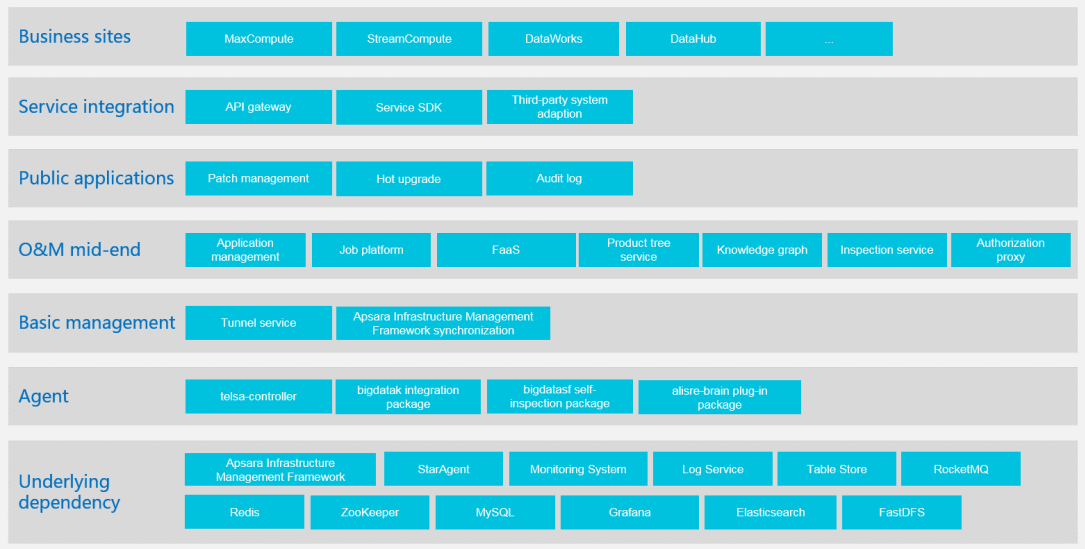
# 32.1.3.1. O&M Architecture

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.

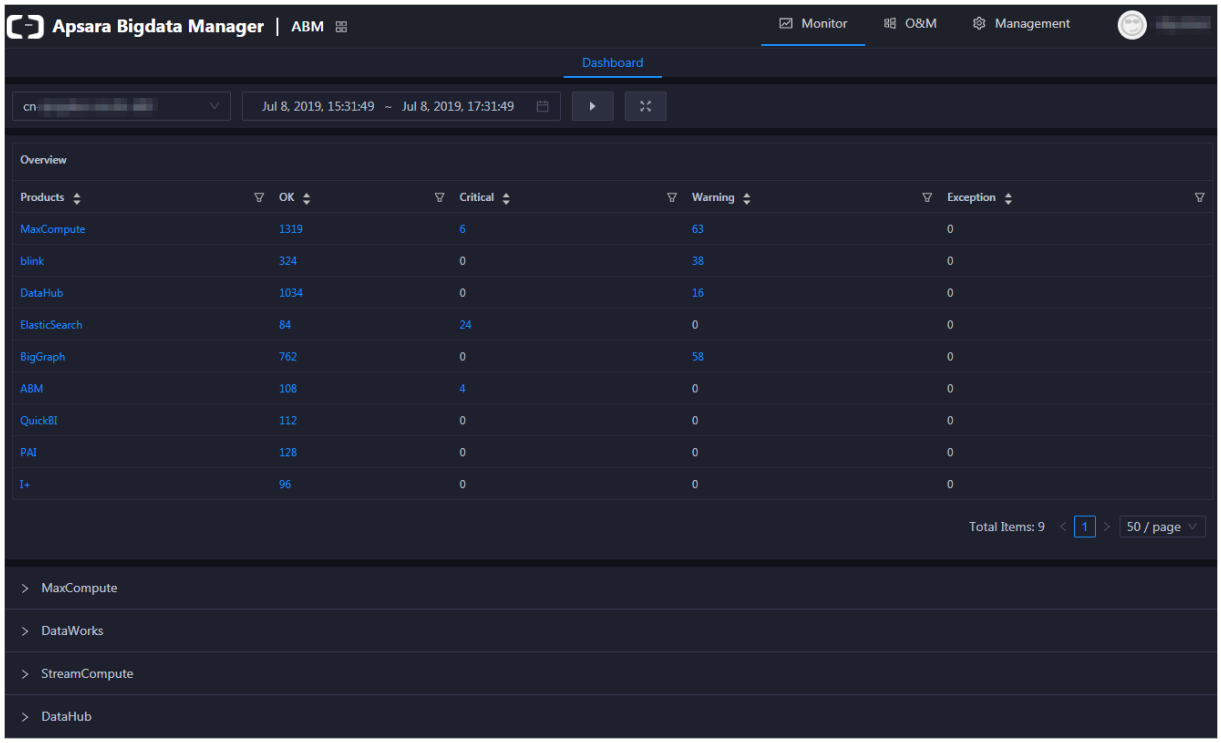
# 32.1.4. Features

## 32.1.4.1. 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

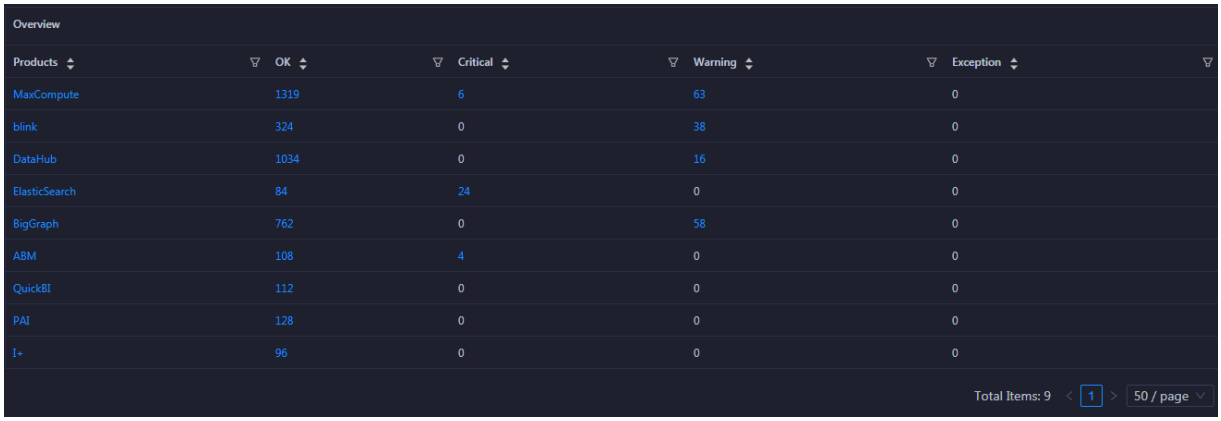
After you log on to the ABM console, the **Dashboard** page appears by default. To return to the **Dashboard** page from another page, click the icon in the upper-left corner and select **ABM**.



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.




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

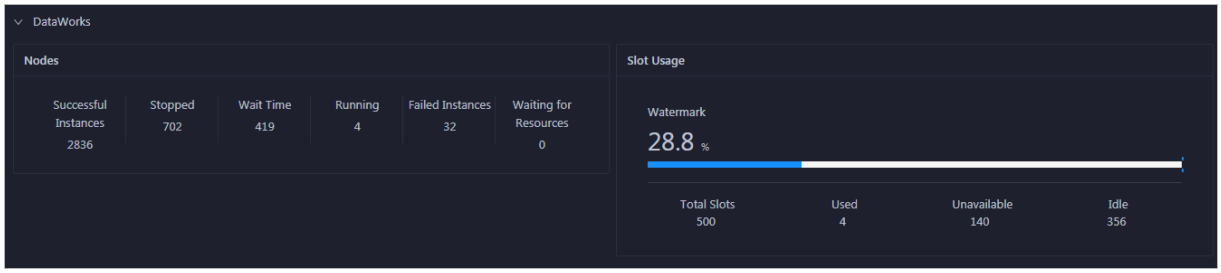
The Dashboard page displays key performance metrics of MaxCompute. To view these metrics, click **MaxCompute** in the Overview section of the **Dashboard** page.



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

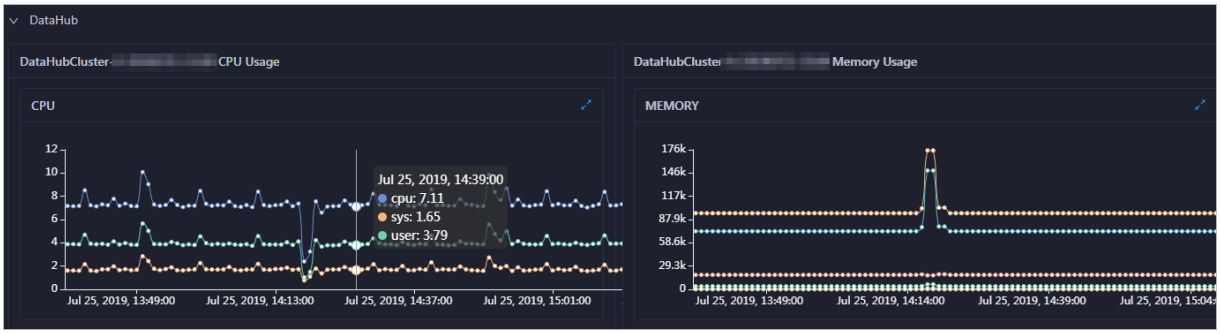
The Dashboard page displays key performance metrics of Realtime Compute for Apache Flink. To view these metrics, click **StreamCompute** in the **Overview** section of the **Dashboard** page.




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.

# 32.1.4.2. Repository

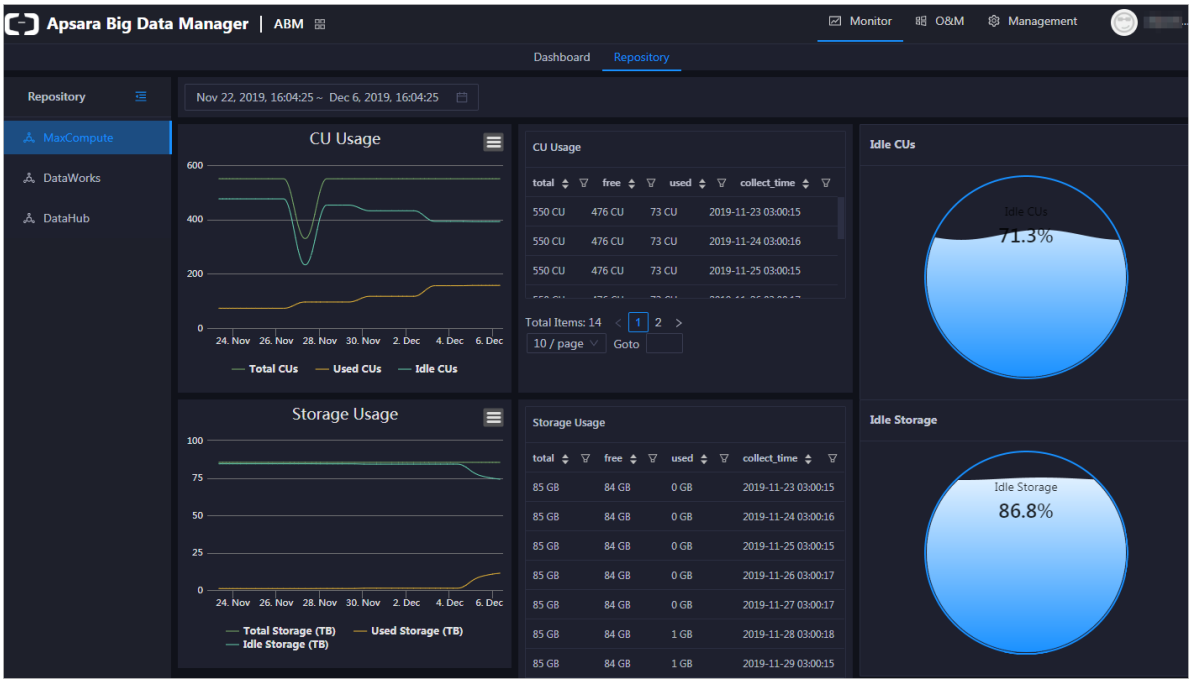
The Repository module displays the resource usage in MaxCompute, DataWorks, and DataHub clusters.

## Repository page

1. Log on to the ABM console. The **Dashboard** page appears by default.

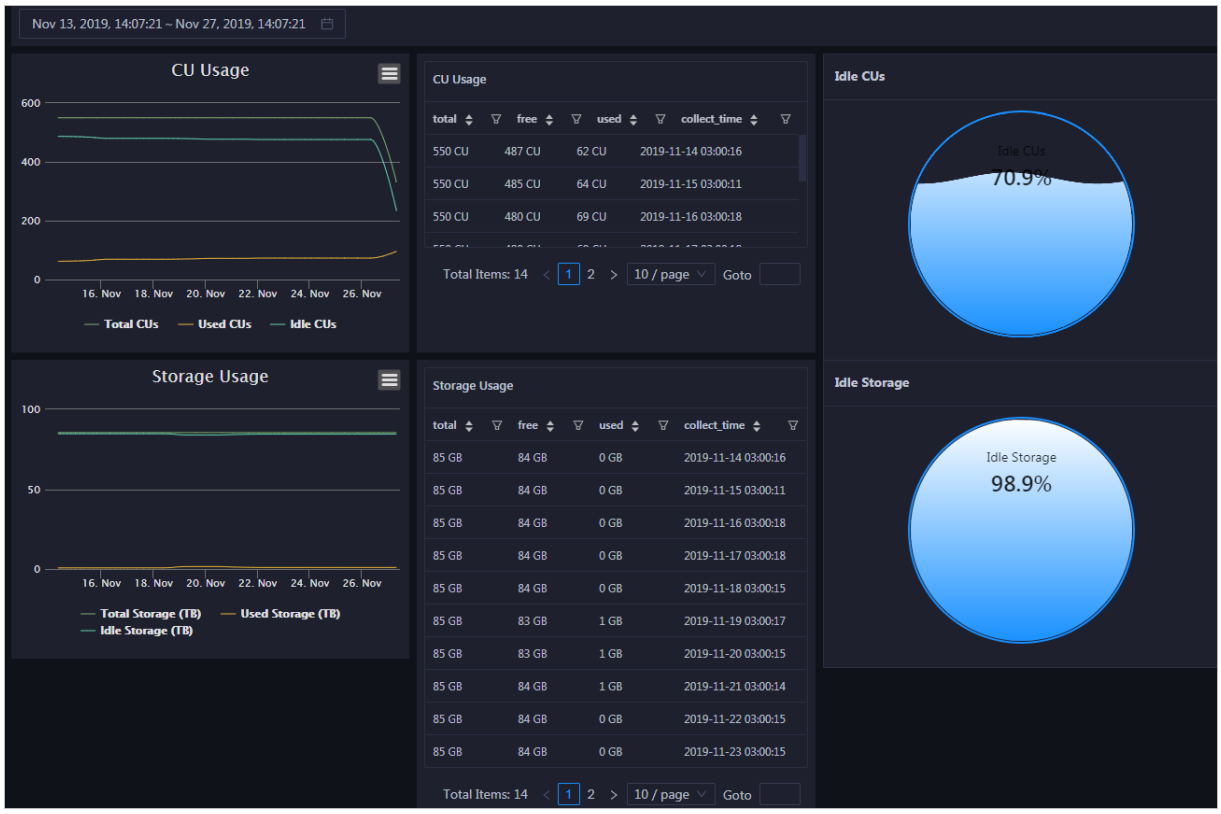
   > **Note** After you log on to the ABM console, the **Dashboard** page appears by default. To return to the **Dashboard** page from any other page, click ≡ in the upper-left corner and select **ABM**.

2. On the **Dashboard** page, click the **Repository** tab to go to the **Repository** page.




## MaxCompute repository

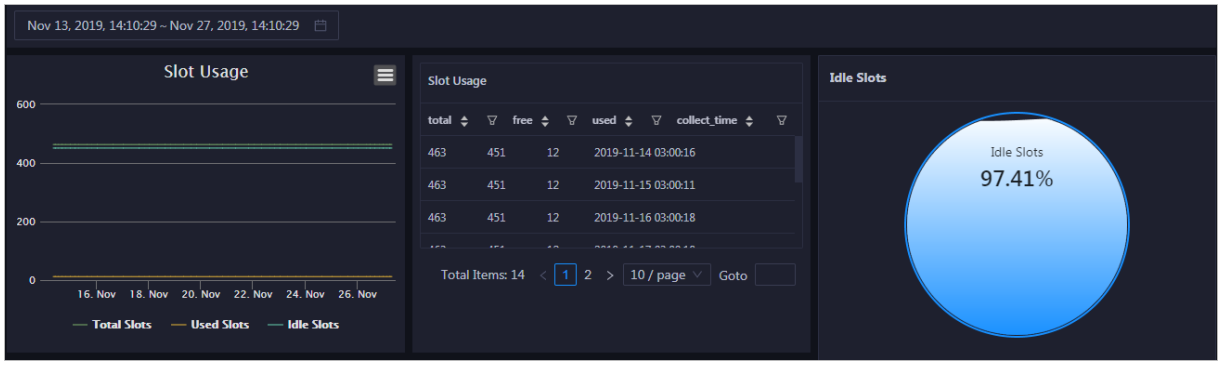
In the left-side navigation pane of the **Repository** page, click **MaxCompute** to view the resource usage in MaxCompute.




This page displays the trends and details of CU and storage usage, and the percentages of idle CUs and storage.

## DataWorks repository

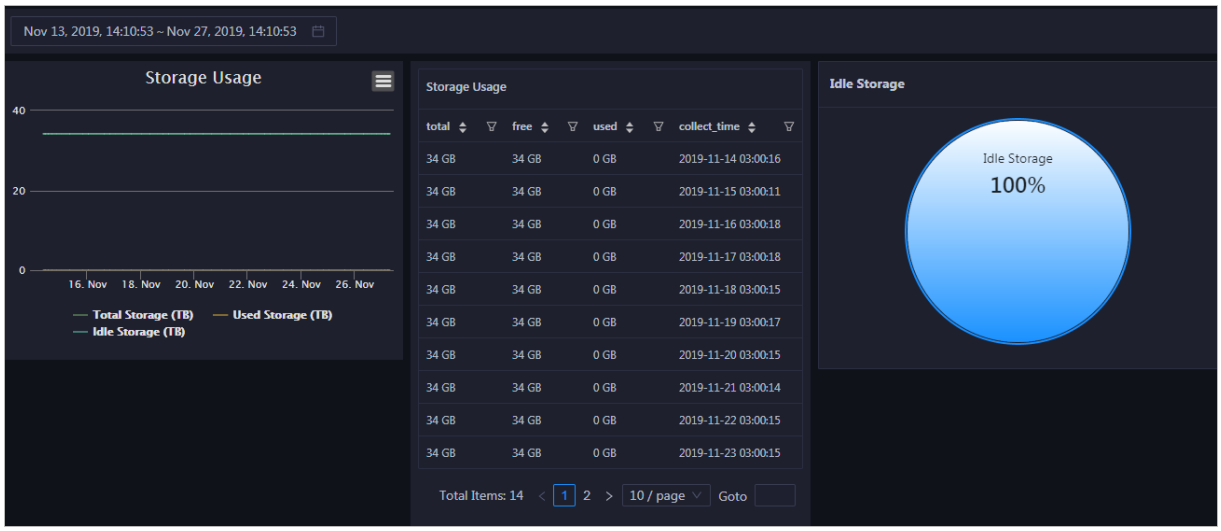
In the left-side navigation pane of the **Repository** page, click **DataWorks** to view the resource usage in DataWorks.




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.

# 32.1.4.3. 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.

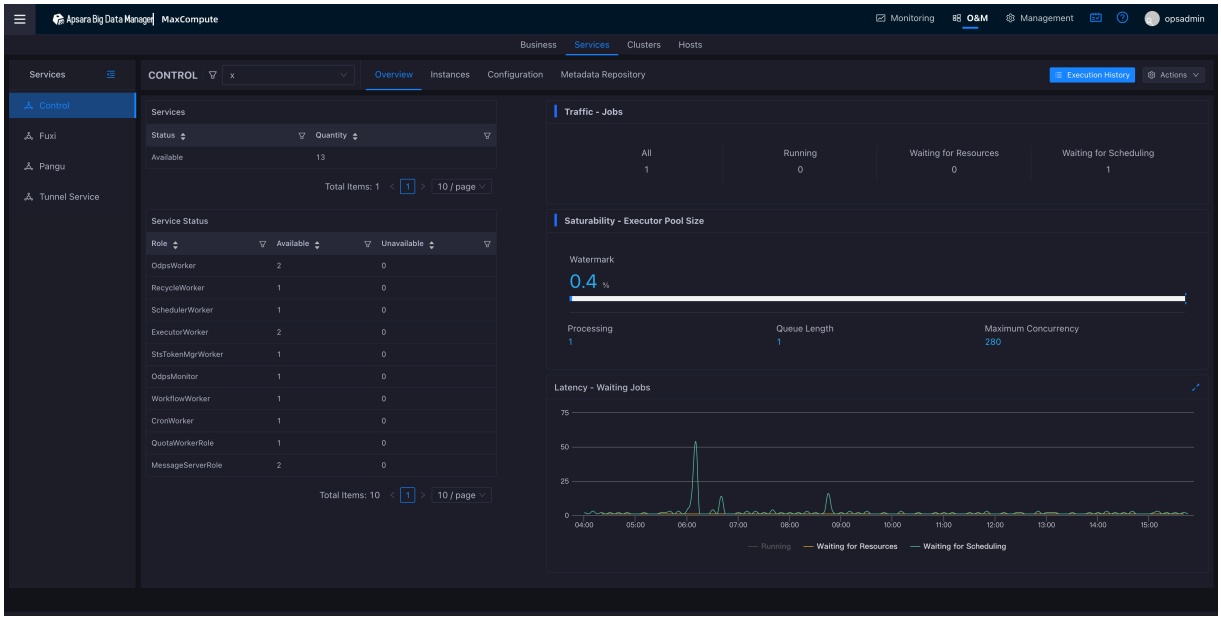
ABM also provides MaxCompute, Realtime Compute for Apache Flink, and DataHub with the following tailored features for cluster O&M:

- 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

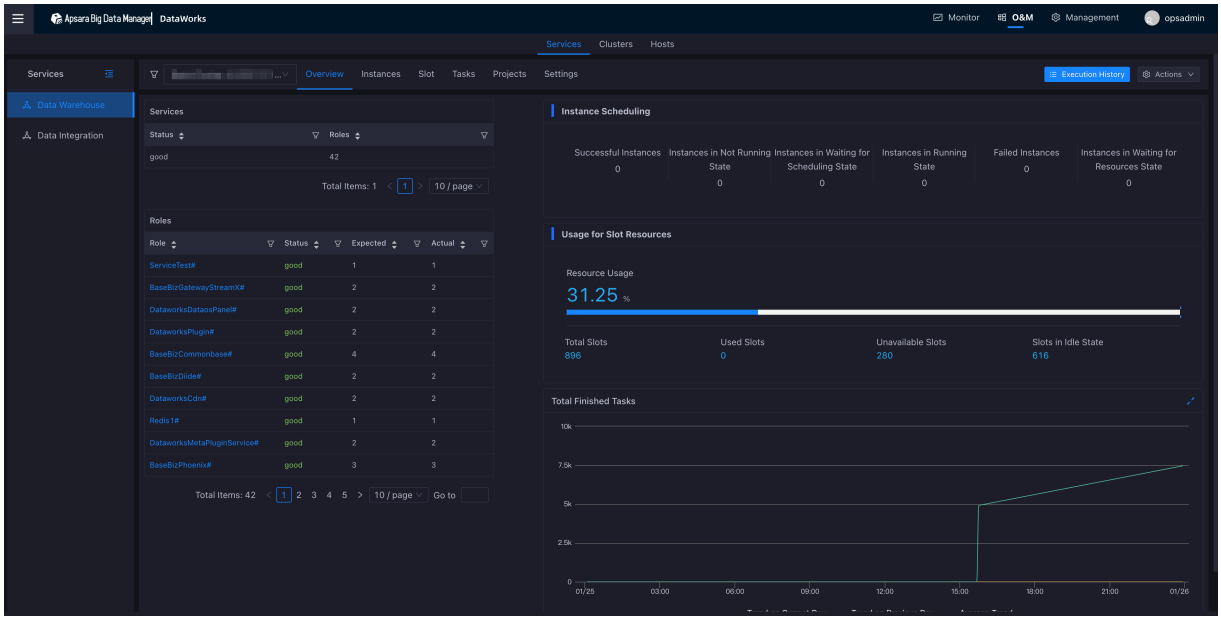
O&M on services is provided for all big data services. ABM provides MaxCompute, DataWorks, DataHub, and Realtime Compute for Apache Flink with tailored features for service O&M.

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, 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, 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, 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.

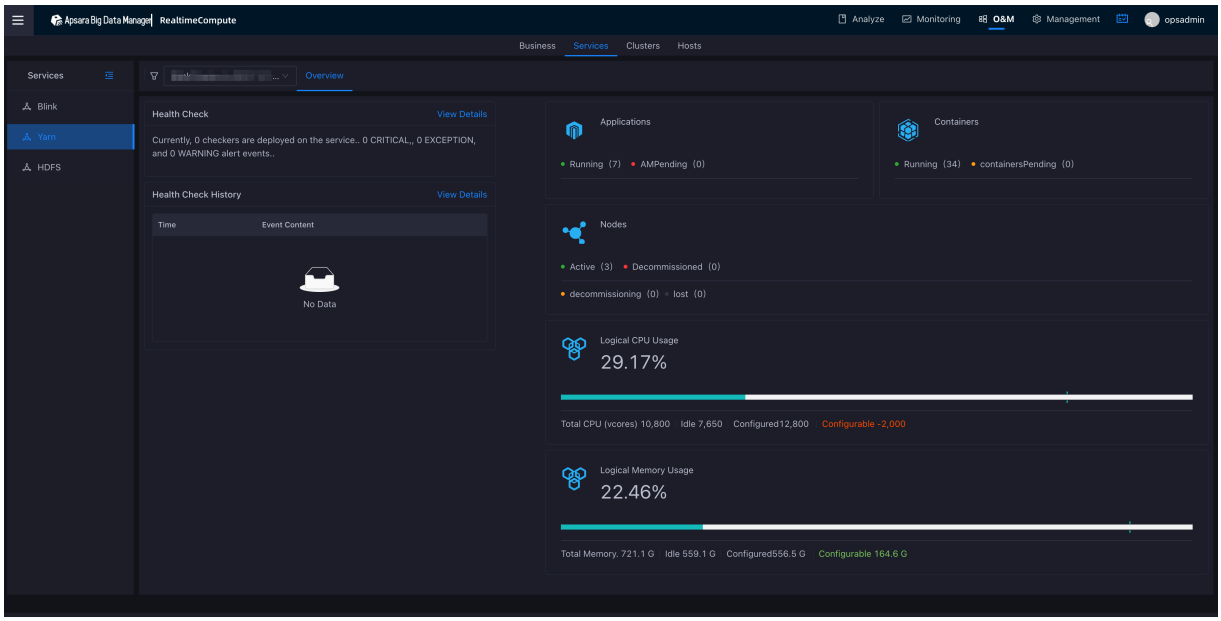
For DataWorks, ABM supports O&M on data warehouses and data integration.




- 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.
- Data Integration: shows general information, integration tasks, and historical analysis.

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).

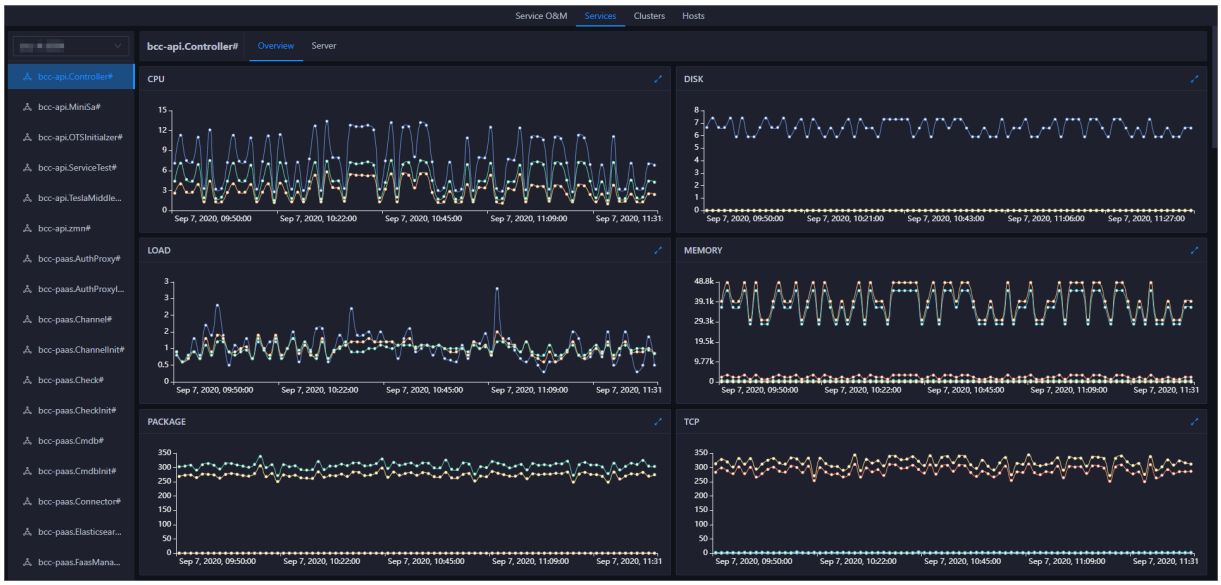



- Realtime Compute for Apache Flink: shows general information about the Realtime Compute service, service status, health check results, and health check history. You can also view the key metrics of clusters, such as Transaction Processing Systems (TPS) and Failover Rate.
- Yarn: shows information about 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.

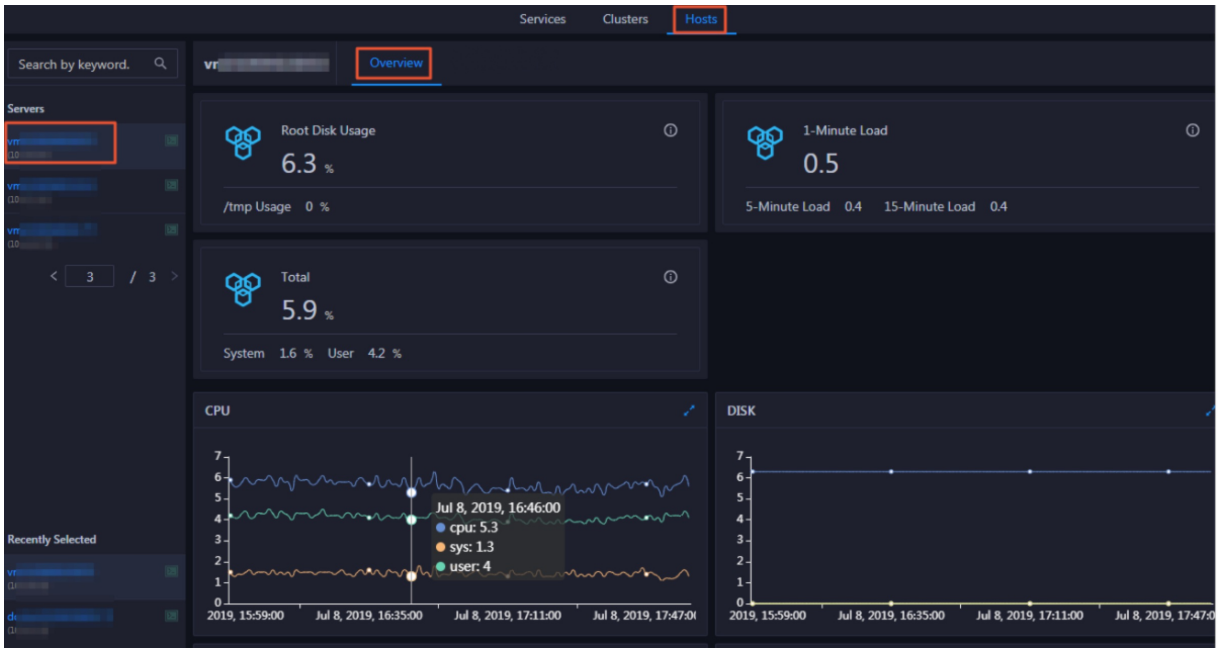
For other big data services, ABM provides information about all server roles in a cluster and the resource usage trend of each server role.




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.

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.

# 32.1.4.4. 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.

## 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.

## 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.

# 32.1.5. 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.

# 32.1.6. Limits

None.

# 32.1.7. 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.