

# Alibaba Cloud Apsara Stack Enterprise Technical Whitepaper

**Version: 1909, Internal: V3.8.1**

**Issue: 20200116**

# 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 due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequent

ial, exemplary, incidental, 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 contact Alibaba Cloud directly if you discover any errors in this document

.





## Document conventions

Style	Description	Example
	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 <b>Danger:</b> Resetting will result in the loss of user configuration data.
	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 <b>Warning:</b> Restarting will cause business interruption. About 10 minutes are required to restart an instance.
	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 <b>Notice:</b> If the weight is set to 0, the server no longer receives new requests.
	A note indicates supplemental instructions, best practices, tips, and other content.	 <b>Note:</b> 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.
<b>Bold</b>	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 <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid Instance_ID</code>
[ ] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>

Style	Description	Example
<b><code>{}</code> or <code>{a b}</code></b>	<b>This format is used for a required value, where only one item can be selected.</b>	<code>switch {active stand}</code>



# Contents

---

<b>Legal disclaimer.....</b>	<b>I</b>
<b>Document conventions.....</b>	<b>I</b>
<b>1 Elastic Compute Service (ECS).....</b>	<b>1</b>
1.1 What is ECS?.....	1
1.2 Architecture.....	3
1.2.1 Overview.....	3
1.2.2 Virtualization platform and distributed storage.....	3
1.2.3 Control system.....	4
1.3 Features.....	5
<b>2 Container Service.....</b>	<b>7</b>
2.1 What is Container Service?.....	7
2.2 Container technology.....	7
2.3 Architecture.....	11
2.4 Features.....	13
<b>3 Auto Scaling (ESS).....</b>	<b>16</b>
3.1 What is ESS?.....	16
3.2 Architecture.....	17
3.3 Features.....	18
3.3.1 Typical scenarios.....	18
3.3.1.1 Overview.....	18
3.3.1.2 Elastic scale-out.....	18
3.3.1.3 Elastic scale-in.....	19
3.3.1.4 Elastic recovery.....	21
3.3.2 Function components.....	21
<b>4 Object Storage Service (OSS).....</b>	<b>23</b>
4.1 Architecture.....	23
4.1.1 System architecture.....	23
4.1.2 Data forwarding procedure.....	25
4.2 Features and principles.....	26
4.2.1 Components.....	26
4.2.1.1 Benefits.....	27
4.2.2 Features.....	29
4.2.3 Terms.....	31
<b>5 Table Store.....</b>	<b>35</b>
5.1 What is Table Store?.....	35
5.1.1 Technical background.....	35
5.1.2 Table Store technologies.....	37
5.2 Benefits.....	38
5.3 Architecture.....	39

5.4 Features.....	41
5.4.1 Users and instances.....	41
5.4.2 Data tables.....	42
5.4.3 Data partitioning.....	43
5.4.4 Common commands and functions.....	43
5.4.5 Authorization and access control.....	44
<b>6 Network Attached Storage (NAS).....</b>	<b>45</b>
6.1 What is NAS?.....	45
6.1.1 Overview.....	45
6.1.2 Benefits.....	45
6.1.3 Scenarios.....	46
6.2 Technical advantages.....	47
6.3 Architecture.....	47
6.4 Features and principles.....	48
6.4.1 Feature overview.....	48
6.4.2 Features.....	49
6.4.3 Terms.....	50
<b>7 Apsara File Storage for HDFS.....</b>	<b>51</b>
7.1 What is Apsara File Storage for HDFS?.....	51
7.1.1 Terms.....	51
7.1.2 Benefits.....	51
7.1.3 Scenarios.....	53
7.2 Design philosophy.....	54
7.3 Architecture.....	55
7.4 Benefits.....	56
<b>8 ApsaraDB for RDS.....</b>	<b>58</b>
8.1 What is ApsaraDB for RDS?.....	58
8.2 Benefits.....	60
8.3 Architecture.....	61
8.4 Features and principles.....	62
8.4.1 Data link service.....	62
8.4.2 High-availability service.....	64
8.4.3 Backup service.....	66
8.4.4 Monitoring service.....	67
8.4.5 Scheduling service.....	69
8.4.6 Migration service.....	69
<b>9 KVStore for Redis.....</b>	<b>70</b>
9.1 What is KVStore for Redis?.....	70
9.1.1 Scenarios.....	70
9.2 Benefits.....	72
9.3 Architectures.....	73
9.3.1 Overall system architecture.....	73
9.3.2 Components.....	75
9.4 Features.....	76

9.4.1 Data link service.....	76
9.4.1.1 Overview.....	76
9.4.1.2 DNS.....	77
9.4.1.3 SLB.....	77
9.4.1.4 Proxy.....	77
9.4.1.5 DB Engine.....	78
9.4.2 HA service.....	78
9.4.2.1 Overview.....	78
9.4.2.2 Detection.....	79
9.4.2.3 Repair.....	79
9.4.2.4 Notice.....	80
9.4.3 Monitoring service.....	80
9.4.3.1 Service-level monitoring.....	80
9.4.3.2 Network-level monitoring.....	80
9.4.3.3 OS-level monitoring.....	80
9.4.3.4 Instance-level monitoring.....	81
9.4.4 Scheduling service.....	81
<b>10 ApsaraDB for MongoDB.....</b>	<b>82</b>
10.1 What is ApsaraDB for MongoDB?.....	82
10.2 Benefits.....	82
10.3 Architecture.....	83
10.4 Features.....	84
10.4.1 Data link service.....	84
10.4.2 High availability service.....	85
10.4.3 Backup service.....	87
10.4.4 Monitoring service.....	88
10.4.5 Scheduling service.....	89
10.4.6 Migration service.....	89
<b>11 KVStore for Memcache.....</b>	<b>91</b>
11.1 What is KVStore for Memcache?.....	91
11.1.1 Scenarios.....	91
11.2 Benefits.....	92
11.3 Architecture.....	93
11.3.1 Overall system architecture.....	93
11.3.2 System components and technical principles.....	95
11.4 Features and principles.....	96
11.4.1 Data link service.....	96
11.4.1.1 Overview.....	96
11.4.1.2 DNS.....	97
11.4.1.3 SLB.....	98
11.4.1.4 Proxy.....	98
11.4.1.5 DB engine.....	98
11.4.2 High availability service.....	99
11.4.2.1 Overview.....	99

11.4.2.2 Detection.....	99
11.4.2.3 Repair.....	100
11.4.2.4 Notice.....	100
11.4.3 Monitoring service.....	100
11.4.3.1 Service-level monitoring.....	100
11.4.3.2 Network-level monitoring.....	100
11.4.3.3 OS-level monitoring.....	101
11.4.3.4 Instance-level monitoring.....	101
11.4.4 Scheduling service.....	101
<b>12 AnalyticDB for PostgreSQL.....</b>	<b>102</b>
12.1 What is AnalyticDB for PostgreSQL?.....	102
12.1.1 Scenarios.....	102
12.2 Benefits.....	105
12.3 Architecture.....	106
12.4 Features.....	108
12.4.1 Distributed architecture.....	109
12.4.2 High-performance data analysis.....	110
12.4.3 High-availability service.....	110
12.4.4 Data synchronization and tools.....	110
12.4.5 Data security.....	111
12.4.6 Supported SQL features.....	111
<b>13 Data Transmission Service (DTS).....</b>	<b>115</b>
13.1 What is DTS?.....	115
13.2 Benefits.....	115
13.3 Architecture.....	116
13.4 Environment requirements.....	117
13.5 Features.....	118
13.5.1 Data migration.....	118
13.5.1.1 Data migration.....	118
13.5.1.2 Data sources.....	118
13.5.1.3 Online migration.....	119
13.5.1.4 Migration modes.....	119
13.5.1.5 ETL features.....	119
13.5.1.6 Migration task.....	120
13.5.2 Data synchronization.....	120
13.5.2.1 Overview.....	120
13.5.2.2 Synchronization tasks.....	120
13.5.2.3 Synchronization objects.....	123
13.5.2.4 Advanced features.....	124
13.5.3 Data subscription.....	124
13.5.3.1 Real-time data subscription.....	124
13.5.3.2 Subscription channels and objects.....	124
13.5.3.3 Advanced features.....	126
<b>14 Data Management Service (DMS).....</b>	<b>127</b>

14.1 What is Data Management Service?.....	127
14.1.1 Product value.....	127
14.2 Benefits.....	130
14.3 Architecture.....	131
14.4 Features.....	134
<b>15 Server Load Balancer (SLB).....</b>	<b>136</b>
15.1 What is Server Load Balancer?.....	136
15.2 Architecture.....	137
15.3 Features.....	140
15.4 Benefits.....	141
15.4.1 LVS in Layer-4 SLB.....	141
15.4.2 Tengine in Layer-7 SLB.....	145
<b>16 Virtual Private Cloud (VPC).....</b>	<b>146</b>
16.1 What is VPC?.....	146
16.2 Benefits.....	148
16.3 Architecture.....	148
16.4 Features.....	151
<b>17 Log Service.....</b>	<b>152</b>
17.1 What is Log Service?.....	152
17.1.1 Overview.....	152
17.1.2 Values.....	152
17.2 Benefits.....	152
17.2.1 Features.....	153
17.2.2 Service benefits.....	154
17.3 Architecture.....	154
17.3.1 Components.....	155
17.3.2 System architecture.....	157
<b>18 Apsara Stack Security.....</b>	<b>158</b>
18.1 What is Apsara Stack Security?.....	158
18.2 Advantages.....	158
18.3 Architecture.....	160
18.4 Features.....	162
18.4.1 Apsara Stack Security Standard Edition.....	162
18.4.1.1 Threat Detection Service.....	162
18.4.1.2 Traffic Security Monitoring.....	165
18.4.1.3 Server Guard.....	167
18.4.1.4 Server Intrusion Detection.....	172
18.4.1.5 Web Application Firewall.....	173
18.4.1.6 On-premises security operations services.....	175
18.4.2 Optional security services.....	179
18.4.2.1 DDoS Traffic Scrubbing.....	179
18.4.2.2 Sensitive Data Discovery and Protection.....	181
<b>19 Key Management Service (KMS).....</b>	<b>186</b>



19.1 What is KMS?.....	186
19.2 Architecture.....	186
19.3 Features.....	188
19.3.1 Convenient key management.....	188
19.3.2 Envelope encryption technology.....	188
19.3.3 Secure key storage.....	189
<b>20 Apsara Stack DNS.....</b>	<b>190</b>
20.1 What is Apsara Stack DNS?.....	190
20.2 Benefits.....	190
20.3 Architecture.....	192
20.4 Features.....	193
<b>21 API Gateway.....</b>	<b>194</b>
21.1 What is API Gateway?.....	194
21.2 System architecture.....	195
21.3 Features.....	196
21.3.1 API lifecycle management.....	196
21.3.2 Multi-protocol access.....	197
21.3.3 Application access control.....	199
21.3.4 Full-link signature verification mechanism.....	201
21.3.5 Anti-replay mechanism.....	202
21.3.6 HTTPS communication based on the SSL certificate of the user.....	203
21.3.7 Support for OpenID Connect.....	204
21.3.8 Bidirectional communication.....	204
21.3.9 Automatic generation of SDKs and API documentation.....	207
21.3.10 Parameter cleaning.....	208
21.3.11 Mappings between frontend and backend parameters.....	208
21.3.12 Throttling.....	209
21.3.13 IP address-based access control.....	210
21.3.14 Log analysis.....	211
21.3.15 Publish an API in multiple environments.....	211
21.3.16 Online debugging.....	212
21.3.17 Mock mode.....	213
21.3.18 Swagger file import.....	213
21.4 Benefits.....	214
<b>22 Enterprise Distributed Application Service (EDAS).....</b>	<b>215</b>
22.1 What is EDAS?.....	215
22.2 Architecture.....	217
22.3 Features and principles.....	218
22.3.1 Full compatibility with Apache Tomcat containers.....	218
22.3.2 Application-centric PaaS platform.....	218
22.3.3 Rich distributed services.....	219
22.3.4 Maintenance management and service governance.....	220
22.3.5 Three-dimensional monitoring.....	220

22.4 Performance features.....	221
<b>23 MaxCompute.....</b>	<b>222</b>
23.1 What is MaxCompute?.....	222
23.1.1 Overview.....	222
23.1.2 Features and benefits.....	224
23.1.3 Benefits.....	225
23.1.4 Scenarios.....	227
23.1.5 Service specifications.....	232
23.1.5.1 Software specifications.....	232
23.1.5.1.1 Overview.....	232
23.1.5.1.2 Control and service.....	232
23.1.5.1.3 Data storage.....	233
23.1.5.1.4 Size of a single cluster.....	233
23.1.5.1.5 Projects.....	233
23.1.5.1.6 User management and security and access control.....	234
23.1.5.1.7 Resource management and task scheduling.....	238
23.1.5.1.8 Data tables.....	238
23.1.5.1.9 SQL.....	239
23.1.5.1.9.1 DDL.....	239
23.1.5.1.9.2 DML.....	240
23.1.5.1.9.3 Built-in functions.....	242
23.1.5.1.9.4 User-defined functions.....	242
23.1.5.1.10 MapReduce.....	242
23.1.5.1.10.1 Programming support.....	242
23.1.5.1.10.2 Job size.....	243
23.1.5.1.10.3 Input and output.....	243
23.1.5.1.10.4 MapReduce computing.....	244
23.1.5.1.11 Graph.....	244
23.1.5.1.11.1 Programming support.....	244
23.1.5.1.11.2 Job size.....	245
23.1.5.1.11.3 Graph loading.....	245
23.1.5.1.11.4 Iterative computing.....	245
23.1.5.1.12 Processing of unstructured data.....	246
23.1.5.1.12.1 Processing of Table Store data.....	246
23.1.5.1.12.2 Processing of OSS data.....	246
23.1.5.1.12.3 Multiple data sources.....	247
23.1.5.1.13 Spark on MaxCompute.....	247
23.1.5.1.13.1 Programming support.....	247
23.1.5.1.13.2 Data sources.....	247
23.1.5.1.13.3 Scalability.....	248
23.1.5.1.14 Elasticsearch on MaxCompute.....	248
23.1.5.1.14.1 Programming support.....	248
23.1.5.1.14.2 System capabilities.....	248
23.1.5.1.15 Other extensions.....	249
23.1.5.2 Hardware specifications.....	249

23.1.5.3 Specifications of DNS resources.....	254
<b>23.2 Architecture.....</b>	<b>256</b>
<b>23.3 Features.....</b>	<b>261</b>
23.3.1 Tunnel.....	261
23.3.1.1 Overview.....	261
23.3.1.2 TableTunnel.....	261
23.3.1.3 UploadSession.....	262
23.3.1.4 DownloadSession.....	264
23.3.2 SQL.....	265
23.3.3 MapReduce.....	265
23.3.4 Graph.....	267
23.3.5 Unstructured data processing (integrated computing scenarios).....	268
23.3.6 Unstructured data processing in MaxCompute.....	268
23.3.7 Enhanced features.....	269
23.3.7.1 Spark on MaxCompute.....	269
23.3.7.1.1 Open-source platform - Cupid.....	269
23.3.7.1.1.1 Overview.....	269
23.3.7.1.1.2 Compatibility with YARN.....	269
23.3.7.1.1.3 Compatibility with FileSystem.....	271
23.3.7.1.1.4 DiskDrive.....	271
23.3.7.1.2 Feature extensions.....	271
23.3.7.1.2.1 Overview.....	271
23.3.7.1.2.2 Security isolation.....	272
23.3.7.1.2.3 Data interconnection.....	272
23.3.7.1.2.4 Client mode.....	272
23.3.7.1.2.5 Spark ecosystem support.....	274
23.3.7.2 Elasticsearch on MaxCompute.....	274
23.3.7.2.1 Terms.....	274
23.3.7.2.2 How Elasticsearch on MaxCompute works.....	276
23.3.7.2.2.1 Overview.....	276
23.3.7.2.2.2 How distributed architecture works.....	276
23.3.7.2.2.3 How full-text retrieval works.....	278
23.3.7.2.2.4 How authentication control works.....	279
<b>24 DataWorks.....</b>	<b>280</b>
24.1 What is DataWorks?.....	280
24.1.1 Product overview.....	280
24.1.2 Scenarios.....	281
24.2 Benefits.....	282
24.3 Architecture.....	284
24.4 Services.....	285
24.4.1 DataStudio.....	285
24.4.2 Data Management.....	286
24.4.3 Data Integration.....	286
24.4.4 Tenant management.....	291

24.4.5 Data Quality.....	291
24.4.5.1 Overview of Data Quality.....	291
24.4.5.2 Use Data Quality to monitor batch data.....	293
24.4.5.3 Use Data Quality to monitor real-time data.....	296
24.4.6 Data Asset Management.....	297
24.4.7 Real-Time Analysis.....	297
24.4.8 Data Service.....	298
24.4.9 Intelligent Monitor.....	298
24.4.10 Scheduling system.....	301
24.4.10.1 Overview.....	301
24.4.10.2 Concepts.....	301
24.4.10.3 Architecture.....	302
24.4.10.4 State machine.....	303
24.4.10.5 Task dependencies.....	304
<b>25 Realtime Compute.....</b>	<b>308</b>
25.1 What is Realtime Compute?.....	308
25.1.1 Background.....	308
25.1.2 Key challenges of Realtime Compute.....	309
25.2 Technical advantages.....	310
25.3 Product architecture.....	314
25.3.1 Business architecture.....	314
25.3.2 Technical architecture.....	315
25.4 Functional principles.....	317
<b>26 DataQ - Smart Tag Service.....</b>	<b>318</b>
26.1 What is DataQ - Smart Tag Service?.....	318
26.1.1 Overview.....	318
26.1.2 Current situation.....	319
26.1.3 Scenarios.....	320
26.1.4 Product benefits.....	321
26.2 Technical benefits.....	321
26.3 Production architecture.....	322
26.4 Features.....	324
26.4.1 Tag center.....	324
26.4.1.1 Overview.....	324
26.4.1.2 Scenarios.....	326
26.4.1.3 Components.....	326
26.4.1.3.1 Tag models.....	326
26.4.1.3.2 Tag warehouse.....	328
26.4.1.3.3 My tags.....	328
26.4.1.3.4 The overview chart.....	328
26.4.1.3.5 Model views.....	328
26.4.1.3.6 Schemas.....	329
26.4.1.3.7 Data import.....	329
26.4.1.4 Technical architecture.....	330

26.4.1.5 Features.....	330
26.4.2 Analysis APIs.....	332
26.4.2.1 Overview.....	332
26.4.2.2 Scenarios.....	332
26.4.2.3 Components.....	334
26.4.2.4 Technical architecture.....	335
26.4.2.5 Features.....	336
26.4.3 Tag factory.....	337
26.4.3.1 Overview.....	337
26.4.3.2 Scenarios.....	337
26.4.3.3 Components.....	338
26.4.3.4 Technical architecture.....	339
26.4.4 Dashboards.....	339
26.4.5 Tag sync.....	339
26.4.6 Homepage.....	340
26.5 Benefits.....	341
<b>27 Apsara Bigdata Manager (ABM).....</b>	<b>342</b>
27.1 What is Apsara Bigdata Manager?.....	342
27.2 Benefits.....	343
27.3 Architecture.....	344
27.3.1 System architecture.....	345
27.4 Features.....	346
27.4.1 Small file merging.....	346
27.4.2 Job snapshot.....	347
<b>28 E-MapReduce (EMR).....</b>	<b>349</b>
28.1 What is EMR?.....	349
28.2 Benefits.....	349
28.3 Architecture.....	350
28.4 Features.....	350
28.4.1 Clusters.....	350
28.4.2 Jobs.....	351
28.4.3 Execution plans.....	351
28.4.4 Alerts.....	351
<b>29 Quick BI.....</b>	<b>352</b>
29.1 What is Quick BI?.....	352
29.2 Benefits.....	352
29.3 Product architecture.....	353
29.3.1 System architecture.....	353
29.3.2 Features.....	354
29.3.3 Deployment.....	356
29.3.4 Server roles.....	357
29.4 Features.....	357
<b>30 Graph Analytics.....</b>	<b>359</b>
30.1 What is Graph Analytics?.....	359

30.2 Benefits.....	360
30.3 Product architecture.....	362
30.3.1 System architecture.....	362
30.3.2 Network architecture.....	364
30.4 Features and principles.....	365
30.4.1 OLEP model.....	365
30.4.2 Data integration.....	366
30.4.3 Separate the graph structure logic from graph details.....	367
30.4.4 Intelligent network.....	368
<b>31 Machine Learning Platform for AI.....</b>	<b>370</b>
31.1 What is machine learning?.....	370
31.2 Benefits.....	372
31.3 Architecture.....	372
31.3.1 System architecture.....	372
31.3.2 Architecture.....	373
31.4 Functions.....	375
31.4.1 Resource allocation and task scheduling.....	375
31.4.2 Model and compilation optimization.....	376
31.4.3 Compute engine.....	378
31.4.4 Online prediction system.....	379
31.4.5 List of functions by module.....	382
31.5 System metrics.....	386
<b>32 Dataphin.....</b>	<b>389</b>
32.1 What is Dataphin?.....	389
32.1.1 About Dataphin.....	389
32.1.2 Features.....	390
32.1.3 Benefits.....	392
32.2 Technical advantages.....	393
32.3 Product architecture.....	395
32.3.1 System architecture.....	395
32.3.2 Technology architecture.....	396
32.4 Features.....	398
32.4.1 Console.....	398
32.4.2 Global design.....	399
32.4.3 Data ingestion.....	400
32.4.4 Data standardization.....	401
32.4.5 Modeling.....	404
32.4.6 Coding.....	405
32.4.7 Resource and function management.....	406
32.4.8 Scheduling and management.....	408
32.4.9 Metadata center.....	410
32.4.10 Data asset management.....	411
32.4.11 Security management.....	412
32.4.12 Adhoc query.....	415

<b>33 Elasticsearch.....</b>	<b>416</b>
33.1 What is Elasticsearch?.....	416
33.2 Benefits.....	417
33.3 Architecture.....	417
33.4 Features.....	419
33.4.1 Kibana console.....	419
33.4.2 Restart an instance.....	419
33.4.3 Refresh.....	420
33.4.4 Basic information.....	421
33.4.5 Cluster upgrade.....	422
33.4.6 Elasticsearch cluster configurations.....	423
33.4.6.1 Security.....	423
33.4.6.2 Word splitting.....	424
33.4.6.3 YML configuration.....	425
33.4.6.3.1 Configuration parameters.....	426
33.4.6.3.2 Custom remote reindexing (whitelisting).....	429
<b>34 DataHub.....</b>	<b>432</b>
34.1 What is DataHub?.....	432
34.1.1 Overview.....	432
34.1.2 Benefits.....	433
34.1.3 Highlights.....	434
34.1.4 Scenarios.....	435
34.2 Architecture.....	436
34.2.1 Feature oriented architecture.....	436
34.2.2 Technical architecture.....	438
34.3 Features.....	439
34.3.1 Data queue.....	439
34.3.2 Checkpoint-based data restoration.....	439
34.3.3 Data synchronization.....	439
34.3.4 Scalability.....	441





# 1 Elastic Compute Service (ECS)

---

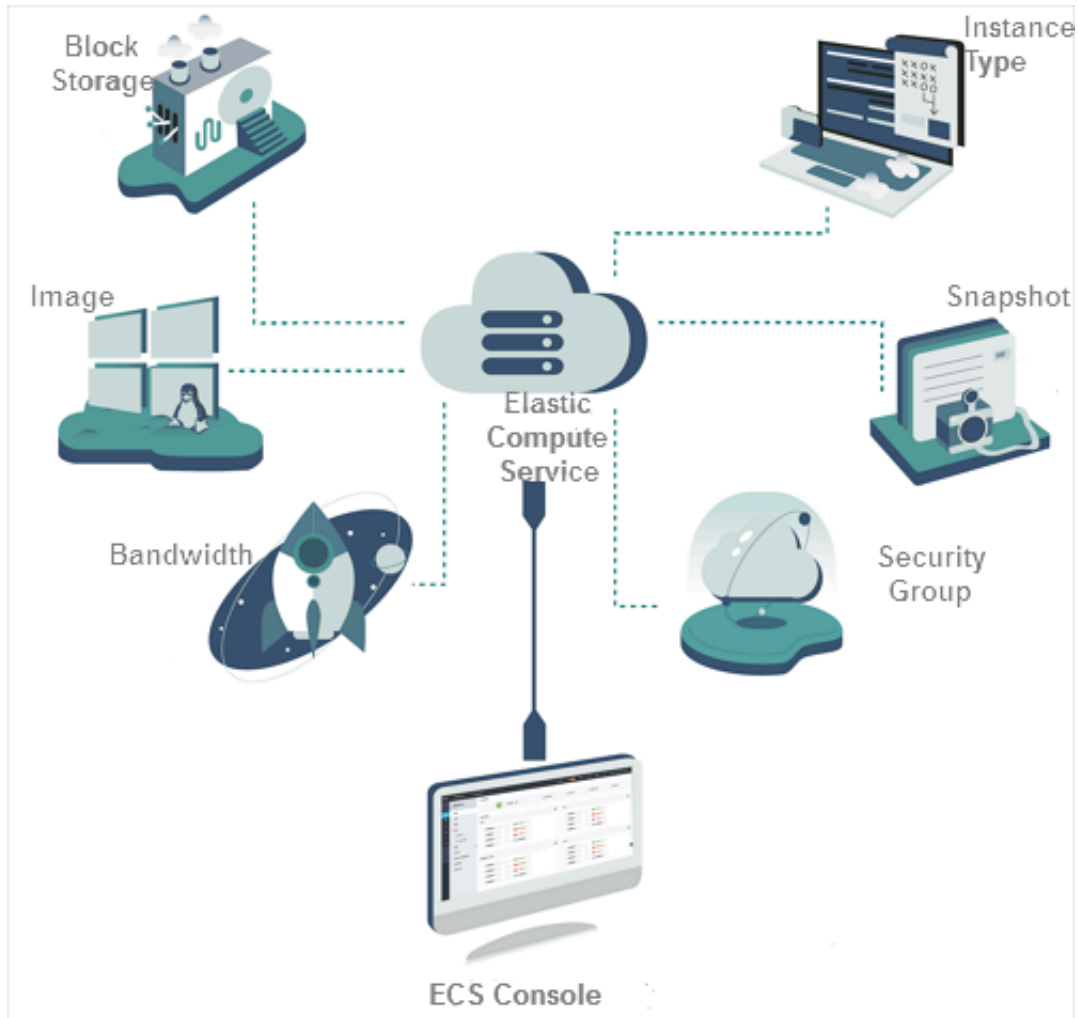
## 1.1 What is ECS?

**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 [Figure 1-1: ECS components](#).

Figure 1-1: ECS components



## 1.2 Architecture

### 1.2.1 Overview

The ECS system is composed of a virtualization platform with distributed storage, a control system, and an O&M and monitoring system.

### 1.2.2 Virtualization platform and distributed storage

The foundation of Elastic Compute Service (ECS) as a service is virtualization.

Apsara Stack uses KVM virtualization to virtualize physical resources and provide them as ECS resources.

ECS contains two important modules: the computing resource module and the storage resource module.

- Computing resources refer to CPU, memory, and bandwidth resources. These resources are created by virtualizing the resources of a physical server and then allocating them to ECS instances for use. The computing resources of a single ECS instances are based on those of a single physical server. When the resources of that physical server are exhausted, you must create a new ECS instance on another physical server to obtain more resources. Resource Quality of Service (QoS) ensures that different ECS instances on a single physical server do not conflict with each other.
- ECS storage is provided by a large-scale distributed storage system. The storage resources of an entire cluster are virtualized and integrated into an external service. The data for a single ECS instance is distributed throughout the entire cluster. In the distributed storage system, all data is saved in triplicate, allowing damaged data in one copy to be automatically replicated from the other copies.

The principles of the virtualization platform and distributed storage are shown in the following figures.

Figure 1-2: Triplicate backup

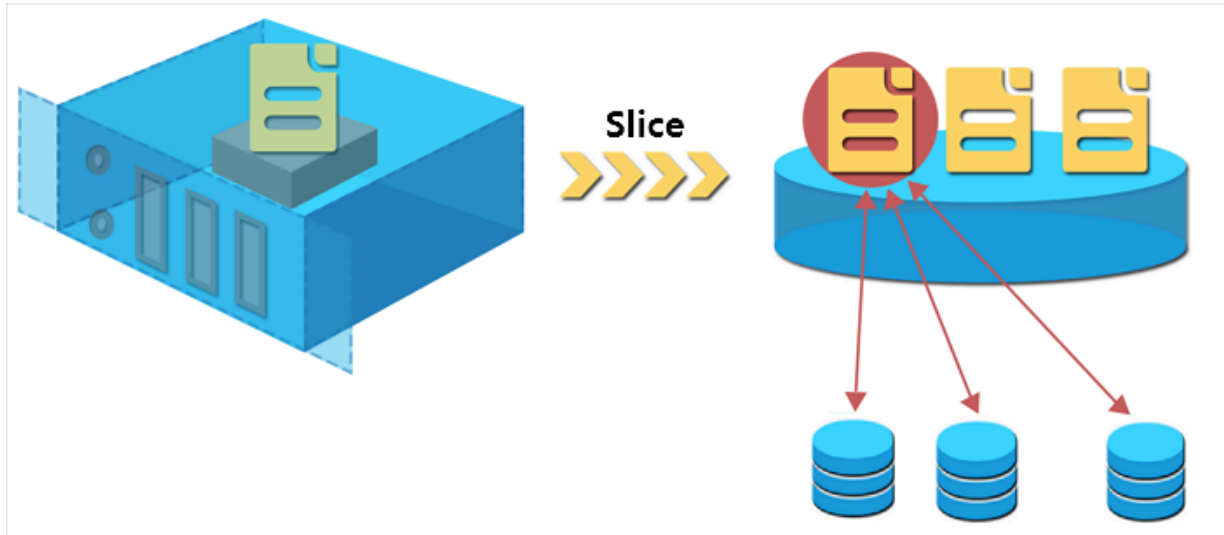


Figure 1-3: Automatic replication



### 1.2.3 Control system

The control system is the core of the ECS platform. It determines which physical servers to start ECS instances on and processes and maintains all ECS functions and information.

The control system is composed of the following modules:

- Data collection module

This module is responsible for data collection throughout the virtualization platform, including computing, storage, and network resource usage. The data collection module serves as the basis for resource scheduling and allows you to centrally monitor and manage cluster resource usage.

- **Resource scheduling system**

**This module determines which physical server to start an ECS instance on. When an ECS instance is created, this module rationally schedules its location based on the resource loads of the physical servers. This module also determines where an ECS instance is restarted when the instance fails.**

- **ECS management module**

**This module can manage and control ECS instances through the start, stop, and restart operations.**

- **Security control module**

**This module monitors and manages the network security of the entire cluster.**

## 1.3 Features

**Instances are the core component that provides computation services to users in Elastic Compute Service (ECS). It only takes a few minutes to create and start an ECS instance. Once an ECS instance is created, it has specific system configurations. ECS instances allow you to compute business data efficiently compared with traditional servers.**

**ECS instances are used and operated in the same way as traditionally-hosted physical servers. You can perform a series of basic operations on ECS instances remotely or through APIs accessed in the control panel.**

**The processing power of ECS instances can be expressed in terms of virtual CPUs and virtual memory, while the storage capabilities of ECS disks are measured by the available capacity of cloud disks. ECS instances support more flexible machine configurations than traditional servers. You can flexibly configure ECS instances as needed if you find that their configurations do not meet your business needs.**

**The lifecycle of an ECS instance begins when it is created and ends when it is released. After an ECS instance is released, all of its data is permanently deleted and cannot be recovered.**

**The ECS console in Apsara Stack console consists of the following tabs:**

- **Overview**

**Provides the number of created and running instances, as well as the distribution of ECS resources in each zone.**

- **Instances**

**You can view and manage the instances you have created on the VMs tab. You can start, stop, restart, and release online instances, as well as log on to a VNC, replace system disks, modify passwords, and change instance configurations. You can also view the basic information and configurations of instances.**

- **Disks**

**You can view and manage the disks you have created on the Disks tab. You can reinitialize disks online, create snapshots, set automatic snapshot policies, release disks, and attach or detach disks. You can also view basic information and mounting information of disks.**

- **Images**

**On the Images tab, you can view, manage, copy, share, and delete images that you have created.**

- **Snapshots**

**On the Snapshots tab, you can view and manage the snapshots you have created. You can roll back disks online, create custom images, and delete snapshots.**

- **Automatic snapshot policies**

**You can view and manage created automatic snapshot policies. You can also set automatic snapshot policies in batches, modify automatic snapshot policy information, and delete automatic snapshot policies.**

- **Security groups**

**On the Security Groups tab, you can view, manage, create, modify, delete, and batch delete security groups, as well as view the instances and rules associated with a security group.**

- **ENIs**

**You can create, modify, delete, view, and manage Elastic Network Interfaces (ENIs), as well as bind or unbind ENIs to ECS instances.**

- **Deployment sets**

**You can create, modify, delete, query, manage, and view the basic information of deployment sets.**

## 2 Container Service

---

### 2.1 What is Container Service?

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

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

### 2.2 Container technology

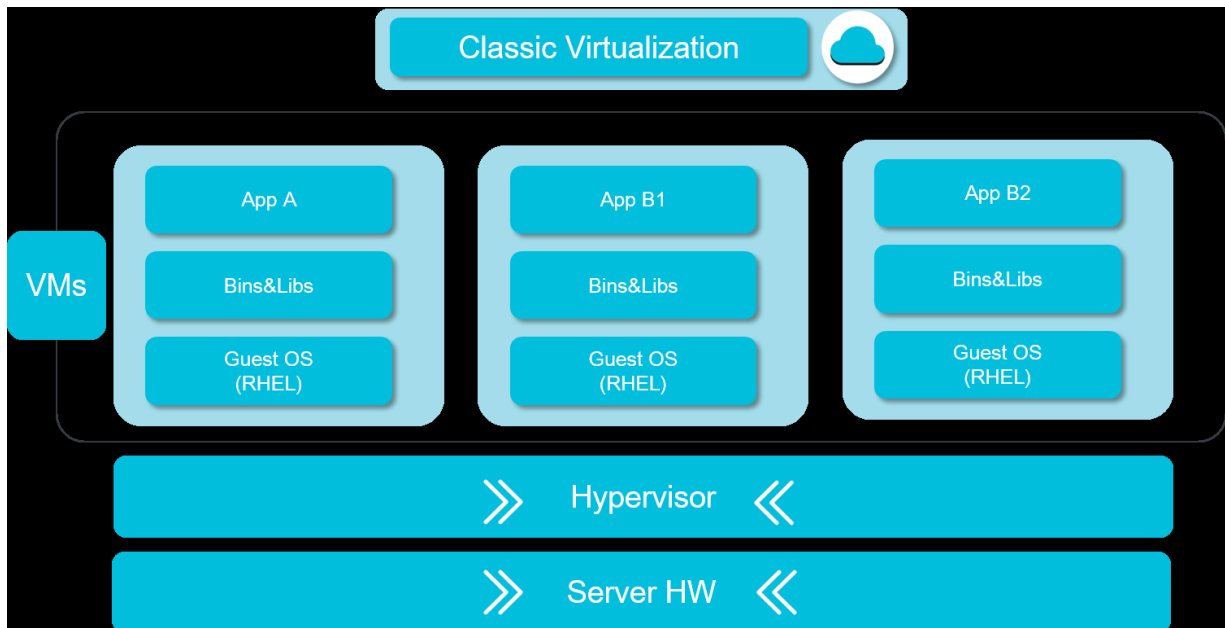
**Containers are a lightweight operating system-level virtualization technology. You can use container images to deliver applications. Container images include applications and their necessary runtime dependencies. Container images have excellent portability and ensure deployment consistency in different environments. Containers are isolated from each other during runtime, ensuring excellent security.**

**Containers avoid potential version conflicts resulting from different applications running in the same environment, and eliminate runtime environment inconsistencies resulting from the same software being run in different environments. Because all containers on a host share the host's OS kernel, containers are more lightweight than virtual machines. This allows you to start containers quickly and gain fine-grained control over container resources.**

## Container technology and virtualization

**Containers do not conflict with conventional virtualization technologies. Conventional virtualization technologies encompass all elements ranging from operating systems to applications, as shown in the following figure.**

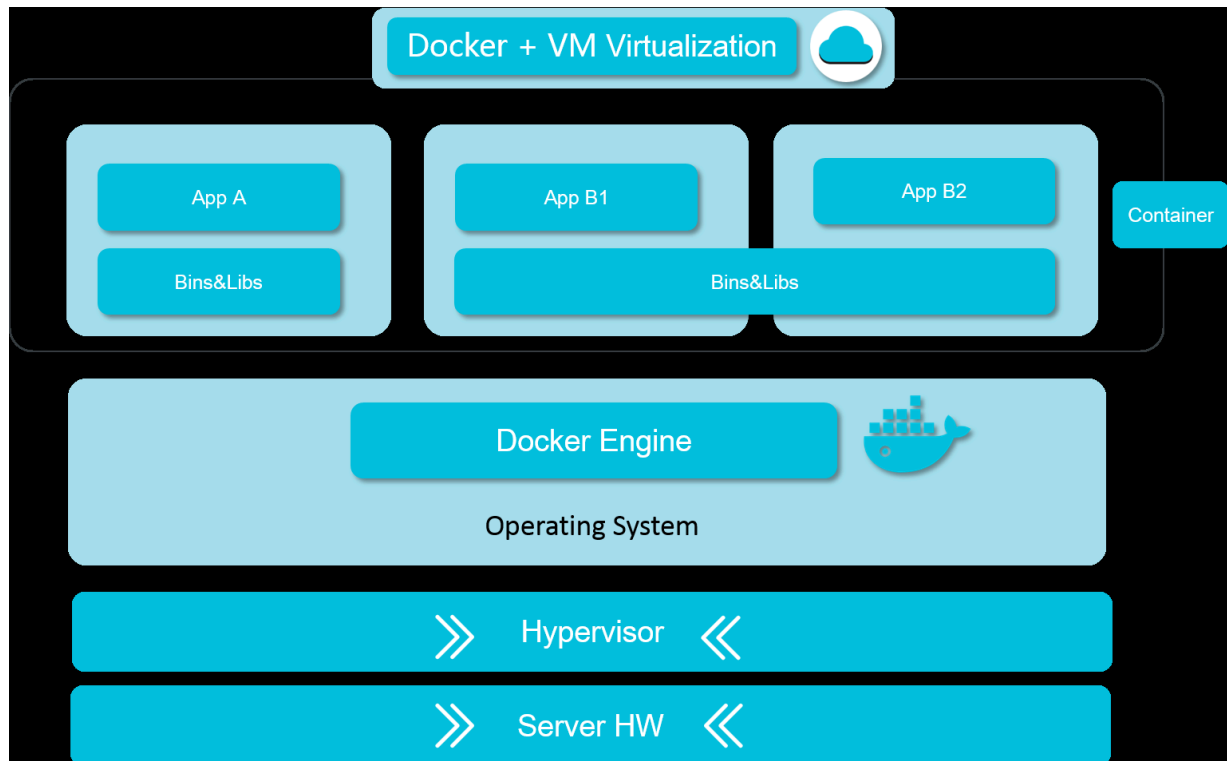
Figure 2-1: Classic virtualization





**Containers only package the application code and its runtime environments. Images can be reused within the same environment in different containers, making containers simple to use and operate.**

Figure 2-2: Combination of Docker and virtualization



**By combining containers and virtualization technologies, you can use virtual machines to provide an elastic infrastructure that offers improved security isolation and live migration capabilities. You can also use the container technology to streamline the deployment and O&M of applications and implement an elastic application architecture.**

#### Technical features

**Containers are agile, portable, and highly-controllable.**

- **Agility:** Containers attract developers with their simplicity and velocity, and allow enterprises to consistently develop and deliver software with greater efficiency.
- **Portability:** Developers can migrate containerized applications from the development environment, to the testing environment, and ultimately to the production environment. During this process, the operating structures for

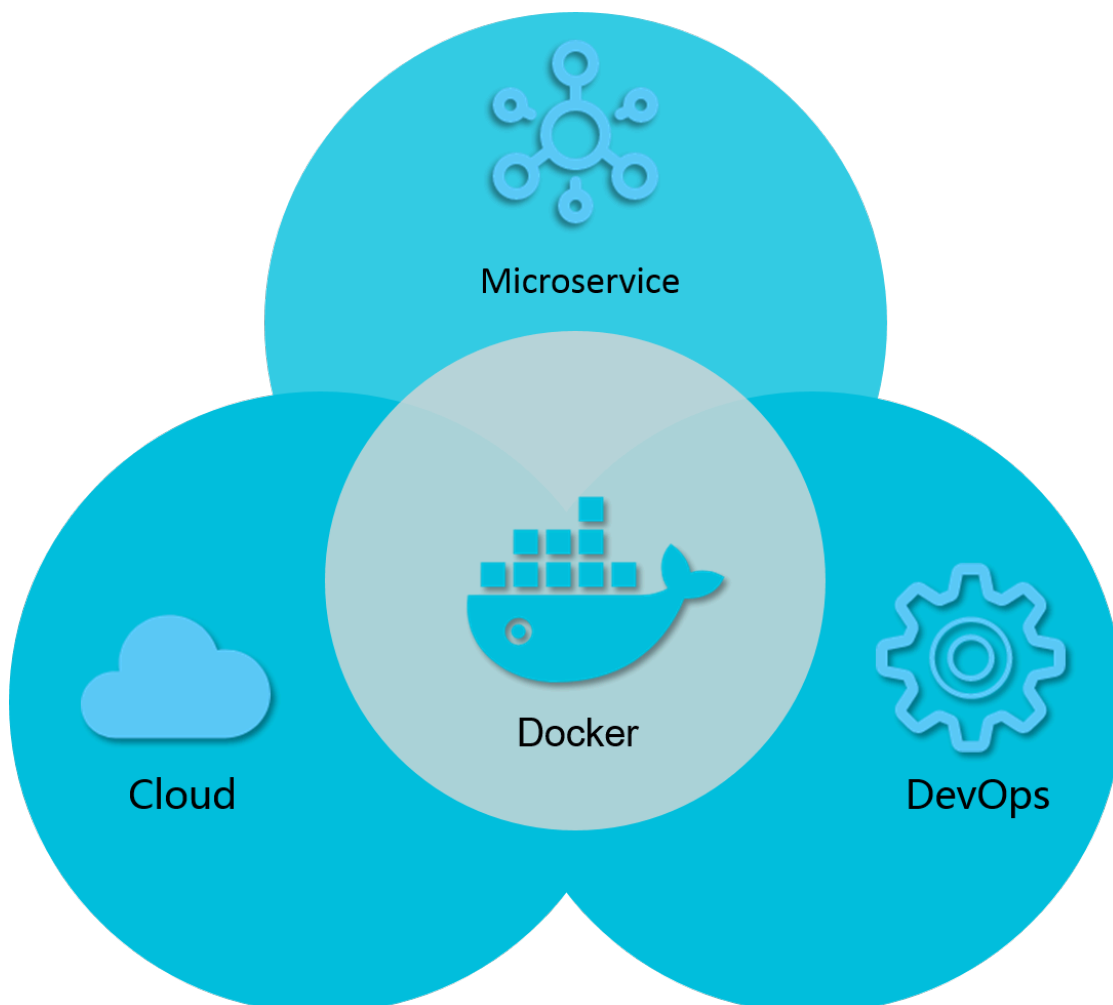
identical images are consistent. Computing capabilities can be deployed across data centers, making computing capability migration a reality in hybrid clouds.

- **Controllability:** Applications in the production environment must meet SLA goals. This requires that you have comprehensive management, security, and monitoring capabilities. Containers provide standardized application environments, allowing developers to use automated tools to manage the infrastructures and applications and ensure that all operations are automated, controllable, and traceable.

## Scenarios

Containers can be applied in a wide range of scenarios. Containers are most often discussed and researched in relation to scenarios that have high container technology requirements, especially DevOps, cloud application management, and microservices.

Figure 2-3: Common scenarios



## 2.3 Architecture

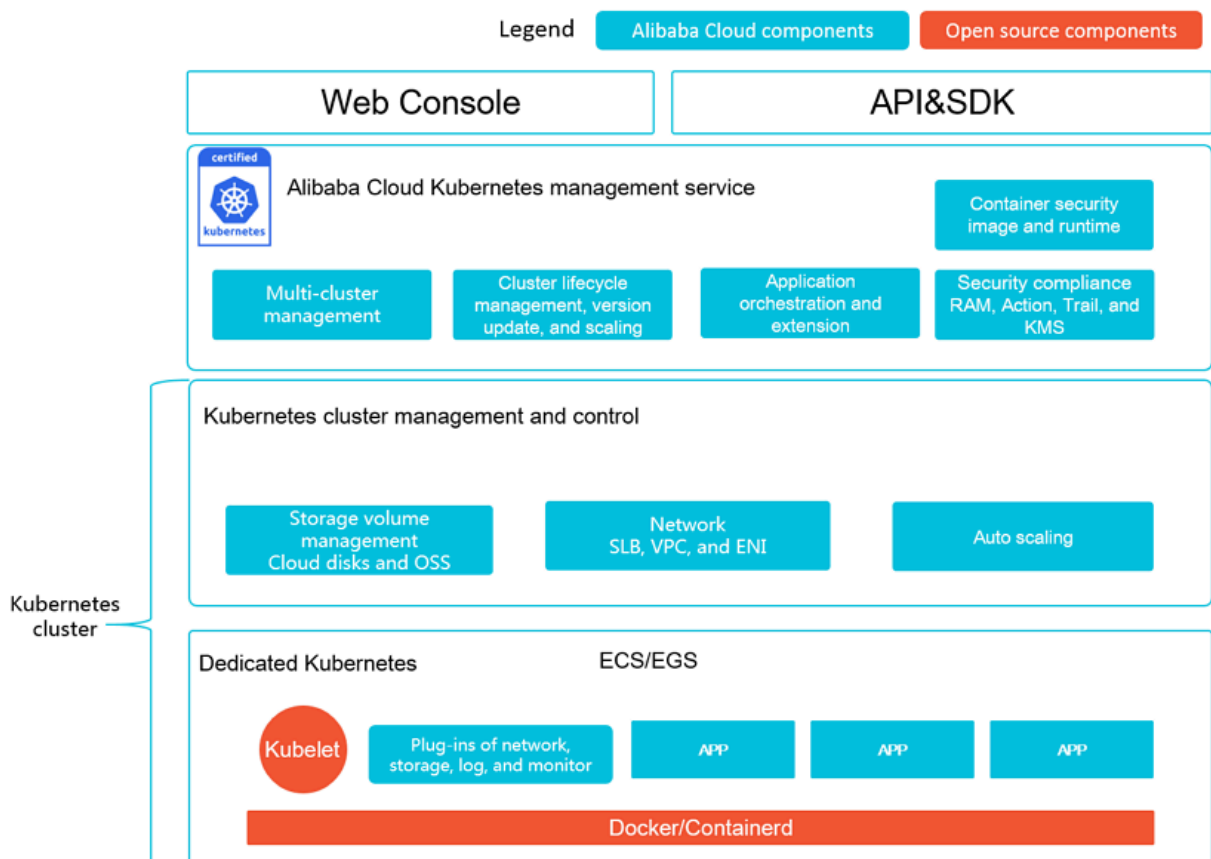
Apsara Stack Container Service supports YAML orchestration and cluster management for Kubernetes to extend and optimize third-party capabilities on Apsara Stack. Container Service allows you to manage clusters and containerized applications through GUIs and APIs.

The underlying architecture allows you to use exclusive cloud servers or physical servers to create a secure and controllable underlying environment where you can customize security group and VPC security rules.

To help migrate your applications to the cloud at a lower cost, Container Service implements APIs that are compatible with standard Docker APIs and all Docker images. Container Service provides Kubernetes YAML orchestration templates which allow you to migrate your applications seamlessly to the cloud. It also provides flexible and customizable mechanisms for third-party capability extensions.

The following figure shows the Container Service architecture.

Figure 2-4: Architecture

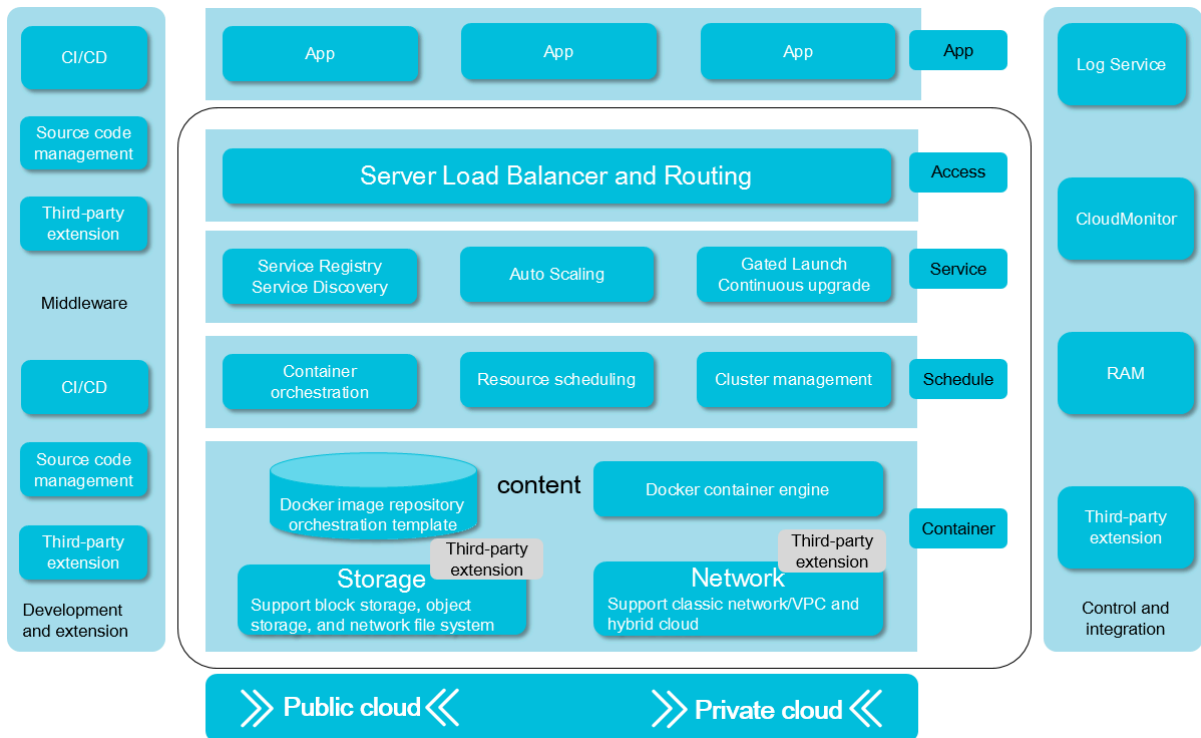


Container Service is adapted and enhanced on the basis of native Kubernetes. This service simplifies cluster creation and scaling and integrates Apsara Stack virtualization, storage, network, and security capabilities, providing the optimal environment to run Kubernetes-based containerized applications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

Feature	Description
Dedicated Kubernetes mode	Integrated with Apsara Stack virtualization technologies, the service allows you to create dedicated Kubernetes clusters . Elastic Compute Service (ECS), Elastic GPU Service (EGS), and ECS Bare Metal instances can be used as cluster nodes . Instances can be flexibly configured to different specifications and support a wide range of plug-ins.
Apsara Stack Kubernetes cluster management and control service	The service provides powerful network , storage, cluster management, scaling, and application extension features.
Apsara Stack Kubernetes management service	The service supports secure images and is highly integrated with Apsara Stack Resource Access Management ( RAM), Key Management Service (KMS ), and logging and monitoring services to provide a secure and compliant Kubernetes solution.
Convenient and efficient use	Container Service for Kubernetes provides services through the Web console, APIs and SDKs.

The following figure shows the Container Service capability stack. Container Service is built on a cloud infrastructure. It is deeply integrated with Apsara Stack capabilities, and supports third-party extensions and applications.

Figure 2-5: Functional architecture



## 2.4 Features

### Features

#### Cluster management

- You can create a classic dedicated kubernetes cluster in 10 minutes on the console, supporting GPU servers.
- Provides OS images for optimizing containers, and the Kubernetes and Docker versions of stability testing and security hardening.
- Supports multi-cluster management and cluster upgrade and scaling.

#### One-stop container lifecycle management

- **Network**

**Provides the high performance Virtual Private Cloud (VPC) and elastic network interfaces (ENI) network plug-in optimized for Alibaba Cloud, which is 20% better than the average network solution.**

**Supports container access policies and flow control restrictions.**

- **Storage**

**Container Service integrates with Alibaba Cloud cloud disk, Network Attached Storage (NAS), and Object Storage Service (OSS), and provides the standard FlexVolume drive.**

**Supports dynamic creation and migration of storage volumes.**

- **The content of the log**

**Supports high-performance automatic log collection and integrating with Alibaba Cloud Log Service.**

**You can also integrate Container Service with third-party open-source log solutions.**

- **Monitoring**

**Supports the monitoring at the level of containers and virtual machines ( VMs). You can also integrate Container Service with third-party open-source monitoring solutions.**

- **Permission**

**Supports Resource Access Management (RAM) authorization and management at the level of clusters.**

**Supports permission configuration management at the level of applications.**

- **Application management**

**Support gray release and blue-green release.**

**Support application monitoring and application elastic scaling.**

**Easily deal with upstream and downstream delivery process by using high-availability scheduling policy**

- **Supports affinity policy and horizontal scaling of services.**
- **Supports high availability across zones and disaster recovery.**

- **Supports the APIs for cluster and application management to easily interconnect with the continuous integration and private deployment system.**

## 3 Auto Scaling (ESS)

---

### 3.1 What is ESS?

**Auto Scaling (ESS) is a management service that automatically adjusts the number of elastic computing resources based on your business demands and strategies. It is suitable for applications with fluctuating business loads, as well as applications with stable business loads.**

**ESS automatically schedules computing resources based on customer strategies and changing business requirements. It provides support for changing business loads and helps control infrastructure costs within an acceptable range. ESS executes scaling based on user-defined scaling policies and modes. When business loads increase, ESS automatically adds ECS instances to ensure sufficient computing capabilities. When business loads decrease, ESS automatically removes ECS instances to save costs. It also replaces unhealthy ECS instances to ensure service performance and safeguard your business.**

**In addition, ESS is seamlessly integrated with Server Load Balancer (SLB) and ApsaraDB for Relational Database Service (RDS). This allows ESS to add or remove ECS instances to or from an SLB backend server group, as well as to add or remove IP addresses of ECS instances to or from an RDS whitelist. ESS eliminates the need to manually perform O&M operations, as it adapts to various complex scenarios and automatically processes business loads based on actual requirements.**



## 3.2 Architecture

ESS is a system that orchestrates ECS instances and provides services based on basic components such as ECS. The ESS system consists of the trigger, worker, database, and middleware services.

Figure 3-1: Architecture

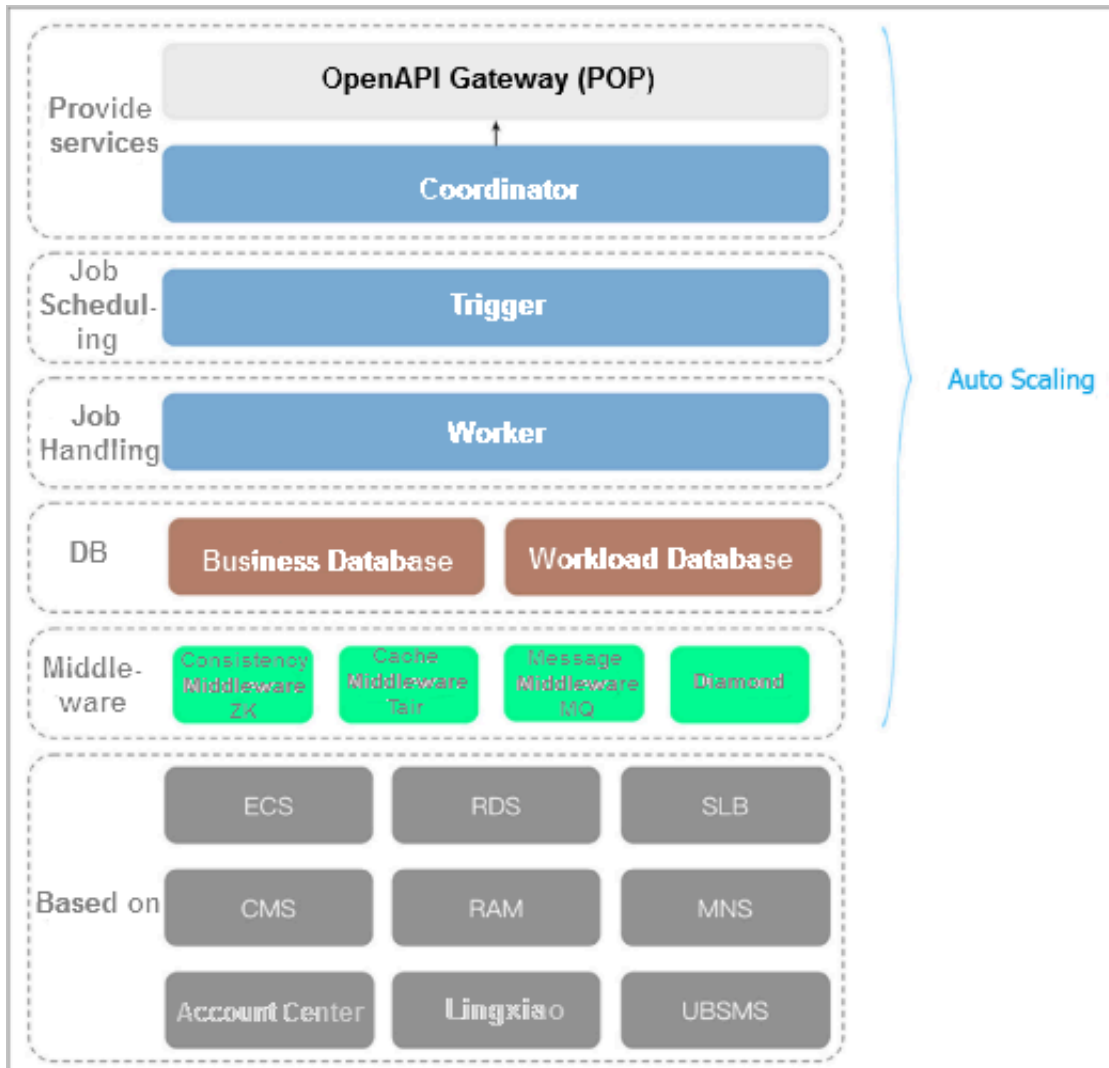


Table 3-1: Architecture description

Layer	Description
Middleware layer	<b>ZooKeeper:</b> ensures consistency by implementing distributed locks for Server Controller.
	<b>Tair:</b> provides caching services for Server Controller
	<b>Message Queue (MQ):</b> provides message queuing services of VM statuses.

Layer	Description
	<b>Diamond:</b> manages persistent configurations.
<b>Database layer:</b> the business database and workload database	<b>Worker:</b> The core of ESS. After receiving a task, it handles the entire life cycle of the task, including splitting, executing, and returning the execution results.
	<b>Trigger:</b> It obtains information from the health checks of instances and scaling groups, scheduled tasks, and CloudMonitor to perform tasks scheduling .
<b>Public-facing services</b>	<b>Coordinator:</b> serves as the ingress of the ESS architecture. It provides external management and control for services, processes API calls, and triggers tasks.
	<b>OpenAPI Gateway:</b> provides basic services such as authentication and parameter passthrough.

### 3.3 Features

#### 3.3.1 Typical scenarios

##### 3.3.1.1 Overview

ESS automatically adjusts the number of elastic computing resources to meet fluctuating business demands. Based on user-defined scaling rules, ESS automatically adds ECS instances as business loads increase to ensure sufficient computing capabilities. When your business loads decrease, ESS automatically removes ECS instances to save costs.

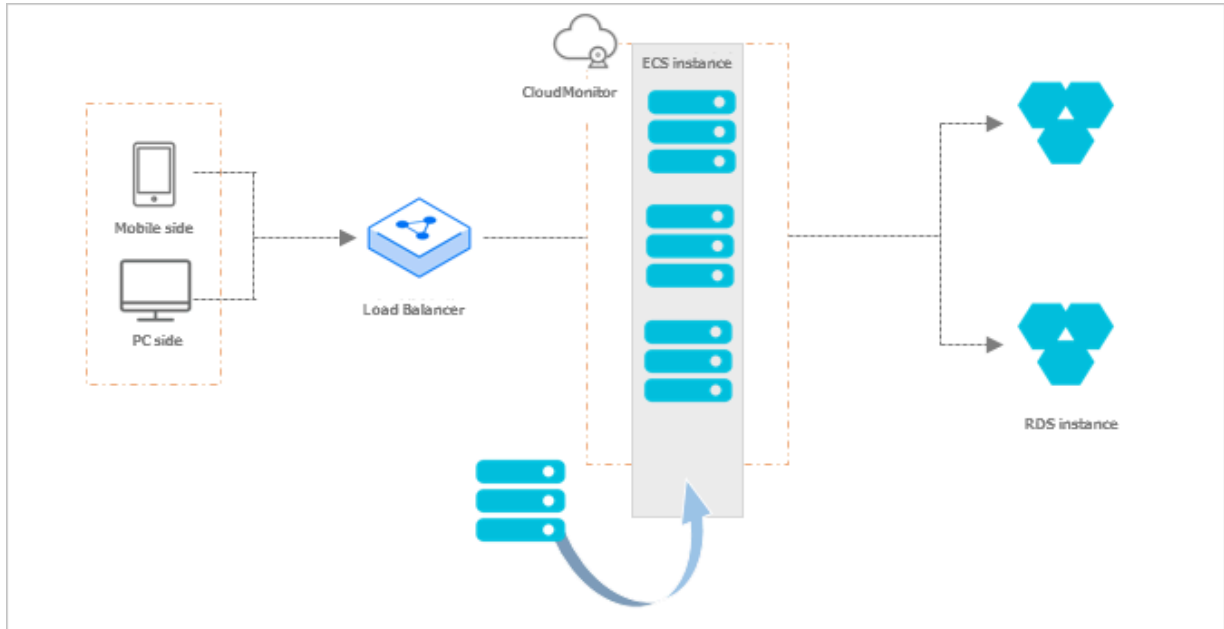
##### 3.3.1.2 Elastic scale-out

When business loads surge, ESS automatically increases underlying resources. This helps maintain access speed and ensure that resources are not overloaded.

You can create scheduled tasks to perform automatic scale-out at specified times or configure CloudMonitor to monitor ECS instance usage in real time and perform scale-out based on actual requirements. For example, when CloudMonitor detects that the vCPU utilization of ECS instances in a scaling group exceeds 80%, ESS elastically scales out ECS resources based on user-defined scaling rules. During

the scale-out process, ESS automatically creates ECS instances and adds these ECS instances to the SLB instance and RDS whitelist.

Figure 3-2: Elastic scale-out



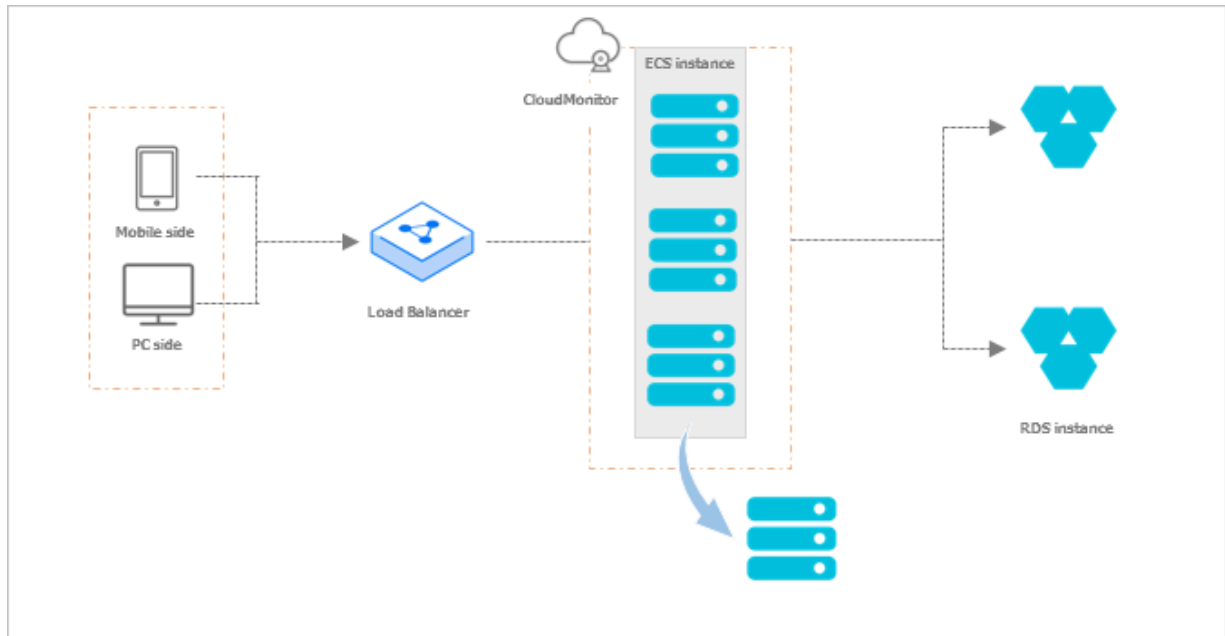
### 3.3.1.3 Elastic scale-in

When loads on services decrease, ESS automatically releases underlying resources to prevent resource wastage and reduce costs.

You can create scheduled tasks to scale in resources automatically at specified points in time. You can also configure CloudMonitor to monitor ECS instance usage in real time and scale in resources based on actual requirements. For example, when CloudMonitor detects that the vCPU utilization of ECS instances in a scaling group falls below a specified threshold, ESS automatically scales in ECS resources based on user-defined rules. During the scale-in process, ESS releases

ECS instances and removes these ECS instances from the SLB instance and RDS whitelist.

Figure 3-3: Elastic scale-in

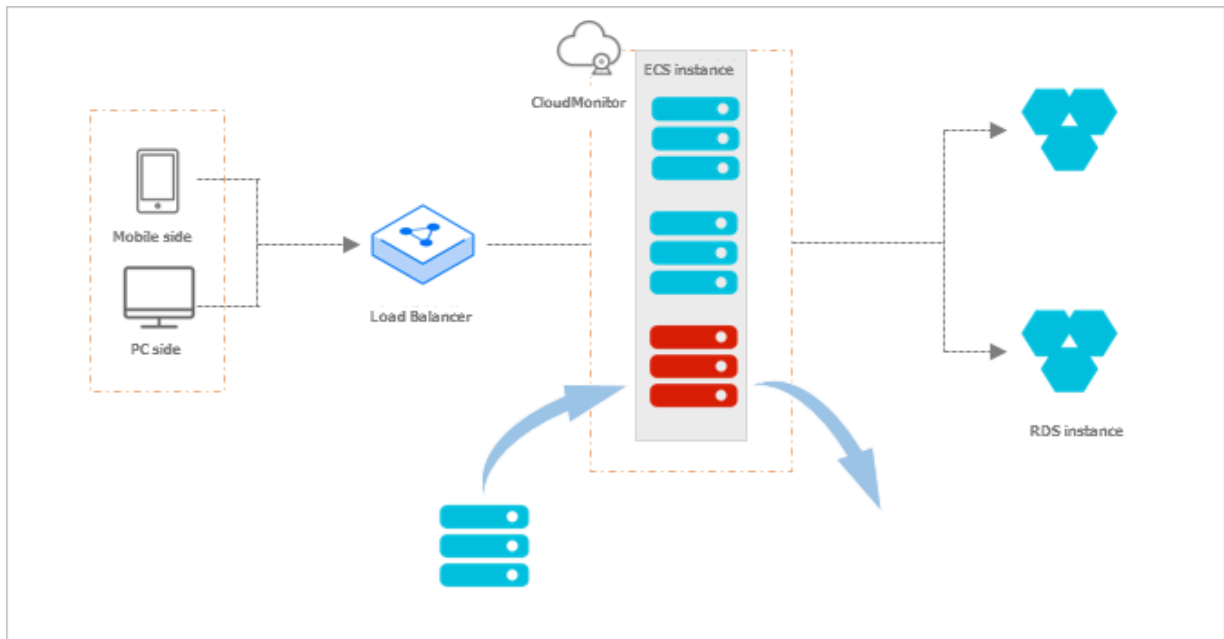


### 3.3.1.4 Elastic recovery

ESS provides a health check function and automatically monitors the health of ECS instances inside scaling groups, so that the number of healthy ECS instances in a scaling group does not fall below the user-defined minimum value.

When ESS detects that an ECS instance is not healthy, it automatically releases the unhealthy ECS instance, creates a new ECS instance, and adds the new instance to the SLB instance and RDS whitelist.

Figure 3-4: Elastic recovery



### 3.3.2 Function components

To create a complete automatic scaling solution that performs scale-in and scale-out based on actual requirements, you need to create scaling groups, configurations, rules, and scheduled tasks.

The following figure shows the procedure to create a complete scaling solution.



## Scaling group

**A scaling group is a group of ECS instances that is dynamically scaled based on the configured scenario. You can specify the maximum and minimum number of ECS instances in a scaling group, as well as the SLB and RDS instances associated with the group.**

## Scaling configuration

**A scaling configuration is a template in ESS for creating ECS instances. When creating a scaling configuration, you can specify ECS instance information, such as instance type, image type, storage size, and instance logon key pair. You can also modify an existing scaling configuration as needed.**

## Scaling rule

**A scaling rule defines the specific scaling activity, for example, the number of ECS instances to be added or removed. The following scaling rules are supported:**

- **Set to N instances:** After this scaling rule is executed, the number of instances in service is changed to N.
- **Add N instances:** After this scaling rule is executed, the number of instances in service is increased by N.
- **Decrease N instances:** After this scaling rule is executed, the number of instances in service is reduced by N.

## Scheduled task

**A scheduled task defines execution actions within a scaling group. It can trigger a specific scaling rule at a specific point in time to execute a scaling activity, such as adjusting the number of ECS instances in a scaling group.**

## 4 Object Storage Service (OSS)

---

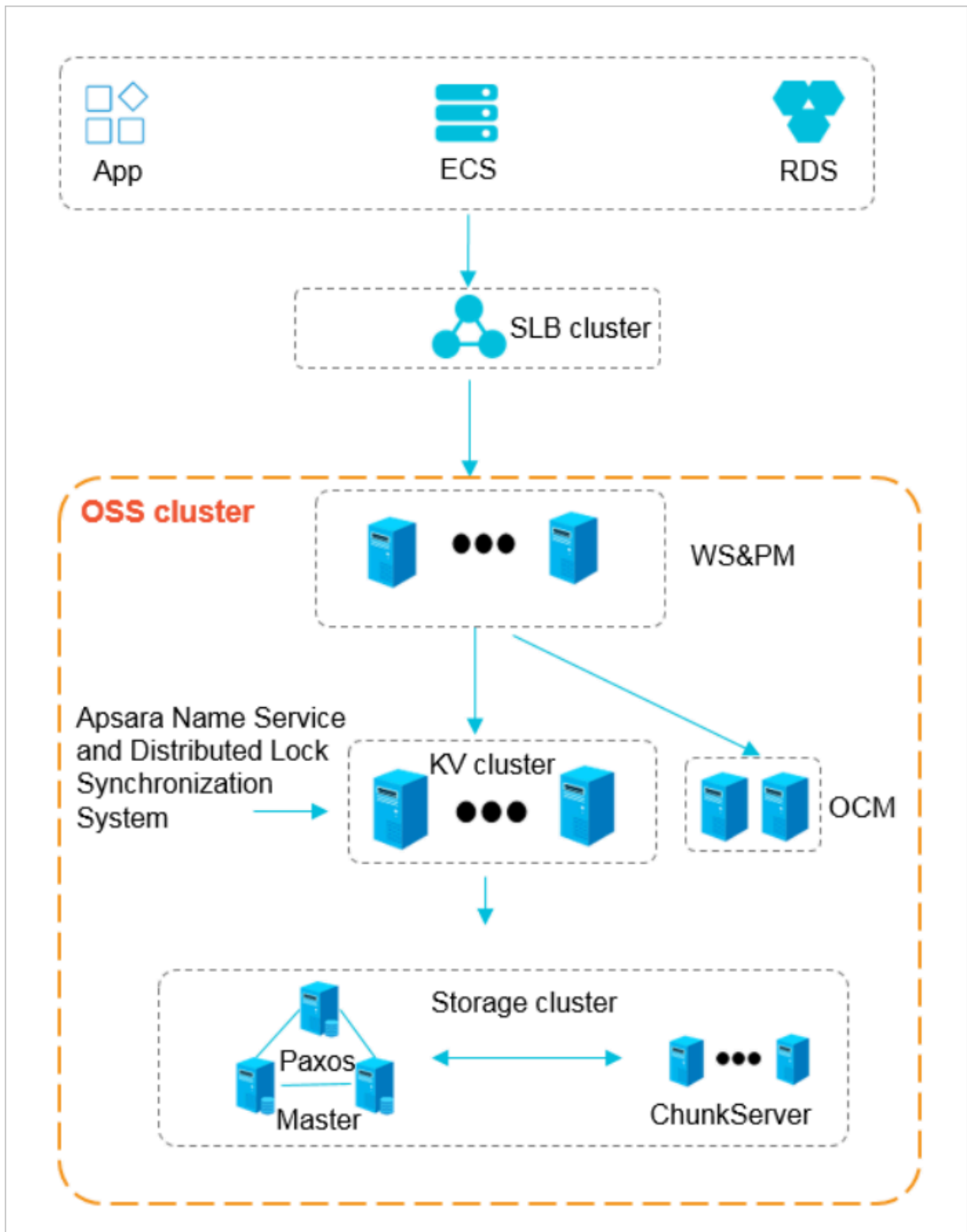
### 4.1 Architecture

#### 4.1.1 System 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. This infrastructure provides OSS and other Alibaba Cloud services with important features such as distributed scheduling, high-speed networks, and distributed storage.**

The following figure shows the architecture of OSS.

Figure 4-1: OSS architecture



The OSS architecture is composed of three layers: protocol access layer, partition layer, and persistent storage layer.



- **Protocol access layer**
  - **WS:** uses the open-source Tengine component, and provides HTTP and HTTPS for external services.
  - **PM:** parses the HTTP request as the read/write operation on the back-end KV or another module. PM also receives and authenticates the user request sent through a RESTful protocol. If the authentication succeeds, the request is forwarded to KV Engine for further processing. If the request fails the authentication, an error message is returned.

- **Partition layer**

The partition layer uses keys to query and store structured data. This layer 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. The partition layer manages indexes of objects, and converts objects to the persistent data objects at the persistent storage layer.

- **SchedulerX** is responsible for naming services and is based on Apsara Name Service and Distributed Lock Synchronization System.
  - **KV** consists of **KVMaster** and **KVServer**. **KVMaster** manages and schedules partitions. **KVServer** stores indexes and actual data of partitions.

- **Persistent layer**

The large-scale distributed file system is deployed at the persistent storage layer. Metadata is stored in 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 faults.

## 4.1.2 Data forwarding procedure

The data forwarding procedure from the perspective of user access is as follows:

User → RESTful API → SLB-Web server (WS) → Protocol module (PM) → KV Engine → Distributed storage

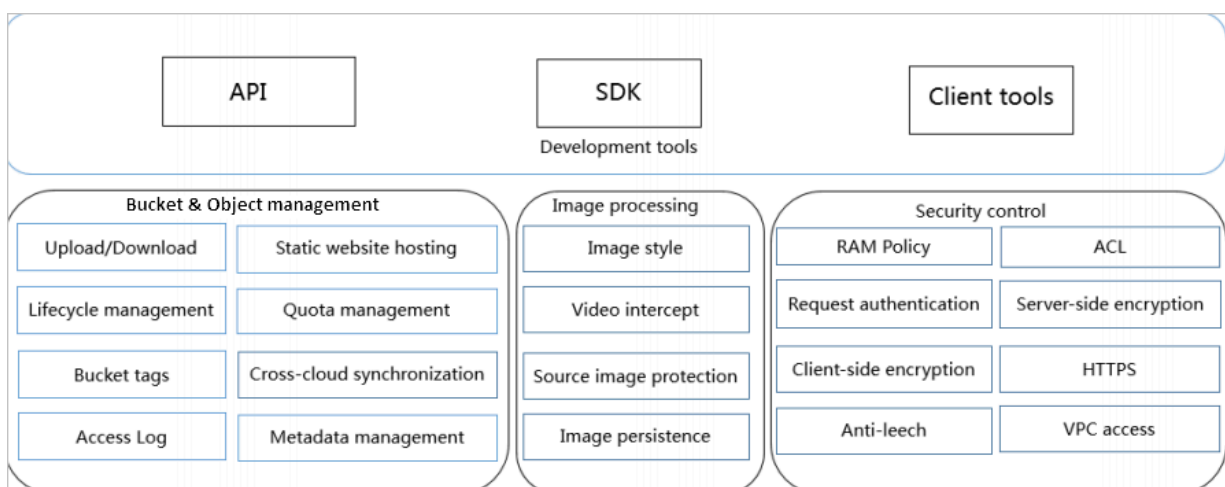
- A user uses different clients such as browsers or SDKs to initiate a request that complies with the convention of the OSS API to the OSS endpoint. The endpoint

parses the request and sends it to the LVS VIP of SLB. The back end of the LVS VIP is bound to the actual WS. The request is forwarded to one of the WSs.

- The PM parses the user request. The specific process is as follows: First, the request is authenticated. If the request fails the authentication, the corresponding error code is returned.
- If the authentication succeeds, the request is parsed as the read/write operation on KV Engine and enters the partition layer.
- The partition layer uses keys to query and store structured data. This layer 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.
- The data stored in KV Engine of the partition layer is written to the persistent storage layer.
- The large-scale distributed file system is deployed at the persistent storage layer. Metadata is stored in 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 software or hardware faults.

## 4.2 Features and principles

### 4.2.1 Components



**OSS is a storage solution that is built on the Apsara system.**

OSS consists of three modules: access layer, application layer, and infrastructure layer.

- **Access layer:** the API, SDKs, and Apsara Stack console
- **Application layer:** bucket and object management, IMG, and security modules
- **Infrastructure layer:** Apsara Distributed File System, Job Scheduler, and Apsara Name Service and Distributed Lock Synchronization System

#### 4.2.1.1 Benefits

##### Multifunctionality

- **Supports multiple functions:** simple upload, form upload, append upload, download, delete, list, and replicate objects, obtain object metadata, and create multipart upload tasks.
- **Supports bucket-based functions:** create, delete, and list objects in a bucket as well as obtain bucket metadata.
- **Creates a globally unique bucket and supports cross-region bucket replication.**
- **Supports lifecycle management,** defines and manages lifecycle rules for all objects in a bucket or a part of an object, and changes capacities and ownership.
- **Supports IMG.** You can obtain image information, convert the image format, resize, crop, and rotate an image, add image, text, and image-text watermarks to an image, customize an image style, call cascade operations in the first-in, first-out (FIFO) order, and protect a source image.
- **Supports zone-disaster recovery.** In the zone-disaster recovery mode, buckets with the same name are replicated. Cluster-based disaster recovery is automatically enabled based on configurations made when the cluster is created. In other words, after a primary bucket is created, a secondary bucket with the same name is created automatically. Information stored in the primary bucket is automatically synchronized to the secondary bucket.
- **Configures static website hosting** for your bucket and allows you to use the bucket domain name to access this static website.
- **Supports hotlinking protection** based on the Referer fields in HTTP headers.
- **Supports cross-origin resource sharing (CORS).** Supports access logging and log analytics in multiple dimensions. You can view access source information.
- **Uses the architecture that features redundancy** to prevent single point of failures (SPOFs).

- **Uploads and downloads large objects, supports multipart upload and range download of large objects, and supports resumable upload, download, and replication.**

#### High performance

**Supports the throughput of a cluster that contains tens of thousands of nodes.**

#### Security

**Supports an ACL for access control. You can configure an ACL when creating a bucket and modify it after it is created. Three levels of permissions are included in an ACL: Private, Public (Read-Only), and Public.**

**Supports Resource Access Management (RAM) for employees, applications, and systems based on the department architecture. A separate logon password or AccessKey pair is created for each employee, application, and system. RAM users do not have any permissions on OSS resources by default. You can use RAM to assign permissions to RAM users or use Security Token Service (STS) for temporary access authorization. HTTPS and traffic encryption on the server and client are supported.**

**Supports the API, SDKs, or migration tools to easily migrate large amounts of data to or from Alibaba Cloud.**

**Supports multiple types of terminals, Web applications, and mobile applications and allows them to write data to or read data from OSS directly. Stream input and object input are supported. You can manage static resources such as images, scripts, and videos on the website in the way you manage folders. After objects are uploaded to OSS, you can apply the features provided by the services in the cloud OS to the objects, such as audio and video processing, IMG, BatchCompute, and offline processing. This way, you can maximize data values.**

**Supports hotlinking protection to prevent unauthorized access.**

**Supports Secure Sockets Layer (SSL) to control the read/write permission on each object.**

**Integrates with the intrusion prevention system to effectively prevent DDoS attacks and CC attacks to ensure that business works properly.**

**Supports cross-region replication to synchronize data to a specified region in real time for geo-disaster recovery. This way, OSS protects important data from extreme disasters and ensures service stability.**

## 4.2.2 Features

### Bucket and object management

- **Bucket overview**

**All buckets of the requester are displayed. If you use HTTP to access the OSS endpoint, all of your buckets are displayed by default.**

- **Create or delete buckets**

- **You can create a maximum of 10 buckets by default. Bucket names must comply with the naming rules.**

**The following scenarios may exist when you create a bucket:**

- **If the bucket you want to create does not exist, the system creates a bucket of a specified name and returns a flag, indicating that the bucket is created.**
- **If the bucket you want to create exists and the requester is the original bucket owner, the original bucket is retained and a flag is returned, indicating that the bucket is created.**
- **If the bucket you want to create exists and the requester is not the original bucket owner, a flag is returned, indicating that the bucket failed to be created.**

**If you want to delete a bucket, ensure that the following conditions are met:**

- **The bucket exists.**
- **You have the permission to delete the bucket.**
- **The bucket contains no objects.**

- **List all objects in a bucket**

**To list all objects in a specified bucket, you must have the corresponding operation permissions on the bucket. If the specified bucket does not exist, an error message is returned.**

**OSS allows you to search for buckets by prefix and set the maximum number (1,000) of objects that can be returned for each search.**

- **Upload or delete objects**

You can upload objects to a specified bucket. You can upload objects to a bucket if the bucket exists and you have the corresponding operation permissions on the bucket. If the object you want to upload has the same name with an object that already exists in the bucket, the new object will overwrite the original object. You can delete a specified object if you have the corresponding operation permissions on the object.

- **Obtain the content or metadata of objects**

To obtain the content or metadata of an object, you must have the corresponding operation permissions on this object.

- **Access objects**

OSS allows you to use a URL to access an object.

#### Image processing (IMG)

- **Custom image styles**

Each change made to an image is represented with a parameter string added to the URL. When many edits are made, the URL becomes very long and unmanageable. IMG allows you to save common operations as an alias (a style). This style feature combines a series of operations into one operation. This style adds only one segment to the URL instead of multiple segments, which shortens the final image URL.

- **Video snapshots**

IMG allows you to process the existing image content and capture the image at a specified point of the video to complete the video snapshot.

- **Source image protection**

To minimize image piracy risks, you must restrict access to the image URLs. Anonymous visitors can obtain only the URLs of thumbnailed or watermarked images. However, source image protection can address this need.

- **IMG persistence**

OSS allows you to perform the SaveAs operation for data processing. This feature enables you to save the processed image to a specified bucket as a resource and assign the image with a specified key. After the image is saved, you can specify the bucket to speed up the resource download when you access the resource

**directly. This feature applies to ultra-large image cropping or other long-latency operations.**

#### Security control

- **Set and query the ACL of a bucket**

**You can set and view the ACL of a bucket. You can set any one of the following permissions for a bucket:**

- **Private:** Authentication is required for users to read from or write to objects in the bucket.
- **Public Read/Write:** Everyone can read files. Authentication is required for users to write to objects in the bucket.
- **Public:** Everyone can read from or write to objects in the bucket.

- **Access logging and monitoring**

**You can choose whether to enable access logging for a bucket. After you enable this feature, OSS pushes access logs on an hourly basis. You can view information such as buckets, traffic, and requests on the Object Storage Service homepage in the Apsara Stack console.**

- **VPC-based access control**

**You can create a single tunnel between OSS and a VPC to access OSS resources over the VPC.**

- **Hotlinking protection**

**OSS provides hotlinking protection to prevent unauthorized domain names from accessing your data in OSS. You can configure the Referer field in the HTTP header to implement hotlinking protection. You can configure a Referer whitelist through the OSS console for a bucket or configure whether to accept access requests where the Referer field is unspecified. For example, you can add `http://www.aliyun.com` to the Referer whitelist for a bucket named `oss-example`. Then, requests with a Referer of `http://www.aliyun.com` can access the objects in the `oss-example` bucket.**

### 4.2.3 Terms

**This topic describes several basic terms used in OSS.**

## object

Files that are stored in OSS. They are the basic unit of data storage in OSS. An object is composed of Object Meta, object content, and a key. An object is uniquely identified by a key in the bucket. Object Meta defines the properties of an object, such as the last modification time and the object size. You can also specify User Meta for the object.

The lifecycle of an object starts when it is uploaded, and ends when it is deleted. Throughout the lifecycle of an object, Object Meta cannot be changed. Unlike the file system, OSS does not allow you to modify objects directly. If you want to modify an object, you must upload a new object with the same name as the existing one to replace it.



### Note:

Unless otherwise stated, objects and files mentioned in OSS documents are collectively called objects.

## bucket

A container that stores objects. Objects must be stored in the bucket they are uploaded to. You can set and modify the properties of a bucket for object access control and lifecycle management. These properties apply to all objects in the bucket. Therefore, you can create different buckets to implement different management functions.

- OSS does not have the hierarchical structure of directories and subfolders as in a file system. All objects belong to their corresponding buckets.
- You can have multiple buckets.
- A bucket name must be globally unique within OSS and cannot be changed after a bucket is created.
- A bucket can contain an unlimited number of objects.

## strong consistency

A feature of operations in OSS. Object operations in OSS are atomic, which indicates that operations are either successful or failed. There are no intermediate states. OSS never writes corrupted or partial data.

Object operations in OSS are strongly consistent. For example, after you receive a successful upload (PUT) response, the object can be read immediately, and the data



is already written in triplicate. Therefore, OSS avoids the situation where no data is obtained when you perform the read-after-write operation. An object also has no intermediate states when you delete the object. After you delete an object, that object no longer exists.

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

#### Comparison between OSS and the file system

OSS is a distributed object storage service that uses a key-value pair format. You can retrieve object content based on unique object names (keys). Although you can use names like test1/test.jpg, this does not necessarily indicate that the object is saved in a directory named test1. In OSS, test1/test.jpg is only a string, which is no different from a.jpg. Therefore, similar resources are consumed when you access objects that have different names.

A file system uses a typical tree index structure. Before accessing a file named test1/test.jpg, you must access directory test1 and then locate test.jpg. This makes it easy for a file system to support folder operations, such as renaming, deleting, and moving directories, because these operations are only directory node operations. System performance depends on the capacity of a single device. The more files and directories that are created in the file system, the more resources are consumed, and the lengthier your process becomes.

You can simulate similar functions in OSS, but this operation is costly. For example, if you want to rename test1 directory test2, the actual OSS operation would be to replace all objects whose names start with test1/ with copies whose names start with test2/. Such an operation would consume a large amount of resources. Therefore, try to avoid such operations when using OSS.

You cannot modify objects stored in OSS. A specific API must be called to append an object, and the generated object is of a different type from that of normally uploaded objects. Even if you only want to modify a single Byte, you must re-upload the entire object. 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 sporadic bursts of access, whereas the performance of a file system is subject to the performance of a single device.

**Therefore, mapping OSS objects to file systems is inefficient, which is not recommended. If attaching OSS as a file system is required, we recommended that you perform only the operations of writing data to new files, deleting files, and reading files. You can make full use of OSS capabilities. For example, you can use OSS to store and process large amounts of unstructured data such as images, videos , and documents.**

## 5 Table Store

---

### 5.1 What is Table Store?

#### 5.1.1 Technical background

Data features in the data technology (DT) era

**As the mobile Internet becomes more common and widely adopted in various industries and fields, Internet applications present the following significant features and trends:**

- **The amount of data that needs to be stored and processed increases exponentially. The data includes microblogs, social events, pictures, and access logs.**
- **With the increase of mobile and IoT devices, the requirements for concurrent writes for structured data storage also increase.**
- **The data has loose schemas and tends to be semi-structured, with data fields that change dynamically.**
- **User access features hot spots and peak hours. For example, during promotional activities, user access soars within a few minutes.**
- **The mobile Internet allows users to connect to Internet applications at any time. Service instability caused by failures (even planned service failures) greatly affects user experience, making high availability a top priority.**
- **Large amounts of data significantly increase the requirements for the performance and scale of compute analysis.**

Challenges of traditional IT software solutions

**Traditional IT software solutions present the following trends and challenges:**

- **Scalability**

**Traditional software, such as relational databases, is incapable of handling such fast-growing data. It bottlenecks data write throughput and access efficiency. With traditional database solutions, the whole process is complex. Databases and tables are partitioned manually and statically. This method requires large amounts of maintenance. In particular scenarios where nodes are added to increase the storage capacity, there is a need to repartition and migrate existing**

data. During this process, it is difficult to guarantee service performance, stability, and availability. The whole process is complex.

- **Data model changes**

Data in traditional databases is processed based on a schema. The number of columns in data is fixed and not changed often. Frequent changes to the table schema and column count affect service availability. Therefore, traditional solutions are incapable of handling the increasing volumes of loosely structured data from Internet applications.

- **Quick scaling**

In traditional solutions, business access loads are stable, and the system is not required to quickly scale resources. When the need to scale resources arises, a large amount of labor is required to reparation and migrate data. Then, when business loads decline, the hosts added during scaling need to be removed to avoid low resource usage, and data needs to be migrated again. This process is extremely complex and inefficient.

- **O&M guarantees**

With traditional software solutions, services are recovered when hardware (network devices or disks) failures occur. Hardware replacement, software upgrades, and configuration tuning and updates need to be performed manually. To ensure that applications are not aware of these processes and avoid deterioration of service availability, users need a special engineering team to achieve system O&M. Therefore, workloads caused from recruitment and fund investment bring a huge challenge to fast-developing enterprises.

- **Computing bottlenecks**

The current business system uses Online Transaction Processing (OLTP) to process and analyze data in relational databases such as MySQL and Microsoft SQL Server. These relational databases are adept at transaction processing. They maintain high consistency and atomicity in data operations, and support frequent data insertion and modification. However, when the volume of data exceeds the processing capabilities of the system, such as when the number of data records reaches tens of millions, or complex calculations are required, OLTP database systems are no longer sufficient.

### 5.1.2 Table Store technologies

**Table Store is a NoSQL data storage service built on the Apsara system that is developed by Alibaba Cloud. Table Store partitions tables and schedules data partitions to different nodes to improve scalability. When a hardware failure occurs, Table Store quickly detects the faulty node by using the heartbeat mechanism and migrates data partitions from the defective node to a healthy node to continue services, achieving rapid service recovery.**

#### Data partitioning and load balancing

**The first primary key column in each row of a table is called the partition key. The system partitions a table into multiple partitions based on the range of the partition key. When the data in a partition exceeds a certain size, the partition is automatically split into two smaller partitions. The data and access loads are distributed to two partitions. The partitions are scheduled to different nodes. Eventually, the linear scalability of the single-table data scale and access loads is achieved.**

**Technical indicator: Table Store can store PBs of data in a single table and allows you to simultaneously read/write millions of data.**

#### Automatic recovery from single point of failures (SPOFs)

**In the storage engine of Table Store, each node serves a number of data partitions in different tables. The Master service role monitors partition distribution and scheduling, and the health of each service node. If a service node fails, the Master service role migrates data partitions from this faulty node to other healthy nodes. The migration is logically performed, and does not involve physical entities, so services can rapidly recover from SPOFs.**

**Technical indicator: SPOFs affect services of some data partitions only and services can recover within several minutes.**

#### Zone-disaster recovery and geo-disaster recovery

**To meet business security and availability requirements, Table Store provides active-standby cluster-based zone-disaster recovery and geo-disaster recovery. Disaster recovery supports instance-based recovery. Any table operation on the primary instance, including insertion, update, or deletion, is synchronized to the table of the same name in the secondary instance. The duration of data synchronization between the primary and secondary instances depends on the network environment of the primary and secondary clusters. In the ideal network**

environment, the synchronization latency is in milliseconds. Before the manual failover, you must stop resource access to the primary cluster and wait for all data to be completely backed up. Do not perform any failover operations in the hour after a recent failover.

In the primary-secondary cluster-based zone-disaster recovery scenario, the endpoints remain unchanged when applications access Table Store in the primary-secondary clusters. In other words, the application endpoints do not need to be changed after the failover. In the primary-secondary cluster-based geo-disaster recovery scenario, the endpoints of the primary-secondary clusters are different. After the failover, endpoints need to be changed for applications.

**Technical indicator:** The RTO of Table Store is less than 2 minutes, the RPO is less than 5 minutes, and the RCO is 1.

## 5.2 Benefits

Table Store is a NoSQL database service that is built on the Apsara system developed by Alibaba Cloud. It enables you to store and access large amounts of structured data in real time. Table Store organizes data into instances and tables, and implements seamless scaling by using data partitioning and load balancing technologies. It shields applications from faults and errors that occur on the underlying hardware platform and provides fast recovery capability and high service availability. Additionally, Table Store manages data with multiple data backups to solid state disks (SSDs), enabling quick data access and high data reliability. As a Table Store user, you only pay for the resources that you reserve and use. You do not have to handle complex issues such as cluster scaling, and upgrades and maintenance of database software and hardware.

Table Store comes with the following features:

- Scalability

There is no upper limit to the amount of data that can be stored in Table Store tables. As data increases, Table Store adjusts partitions to provide more storage space for tables and improve the capability of handling access request bursts.

- Reliability

Table Store provides high data reliability. It stores multiple data replicas and restores data when some replicas become invalid.

- **High availability**

**Table Store 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**

**Table Store automatically performs complex O&M tasks, such as the management of data partitions, software and hardware upgrades, configuration updates, and cluster scale-out.**

- **Access security**

**Table Store provides multiple permission management mechanisms. It performs identity authentication and authorization for each application request to prevent unauthorized data access and secure data access.**

- **High consistency**

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

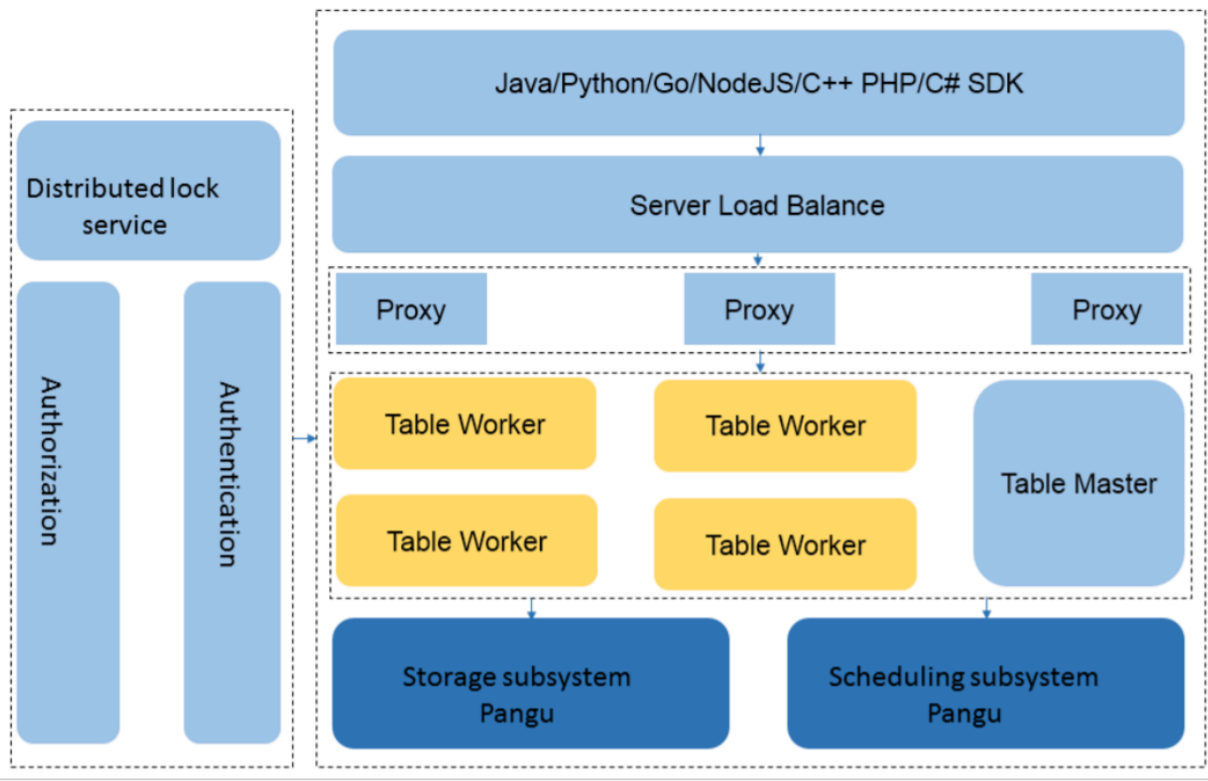
- **Flexible data models**

**Table Store tables do not require a fixed format. Each row can contain a different number of columns. Table Store supports multiple data types, such as integer, boolean, double, string, and binary.**

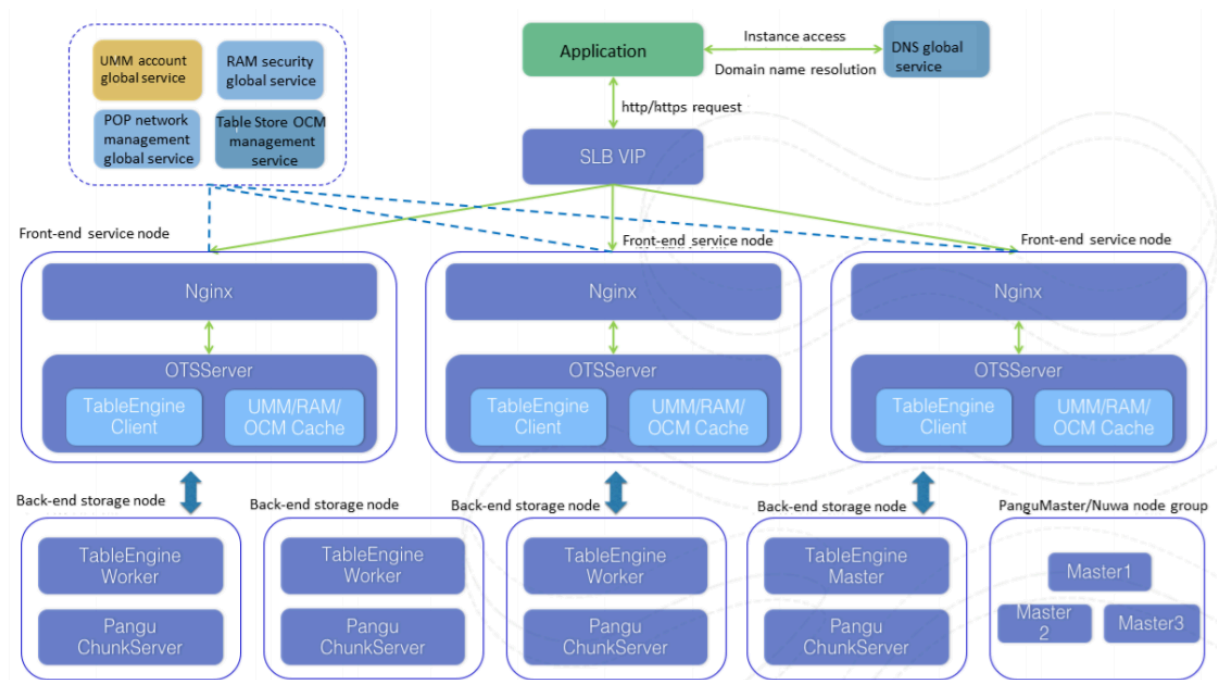
## 5.3 Architecture

**The architecture of Table Store 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 performance writes. 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 Table Store.**



The following figure shows the detailed architecture of Table Store.



- The top layer is the protocol access layer. SLB distributes user requests to various proxy nodes. The proxy nodes receive requests that are sent through 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, an error message is directly returned to the user.

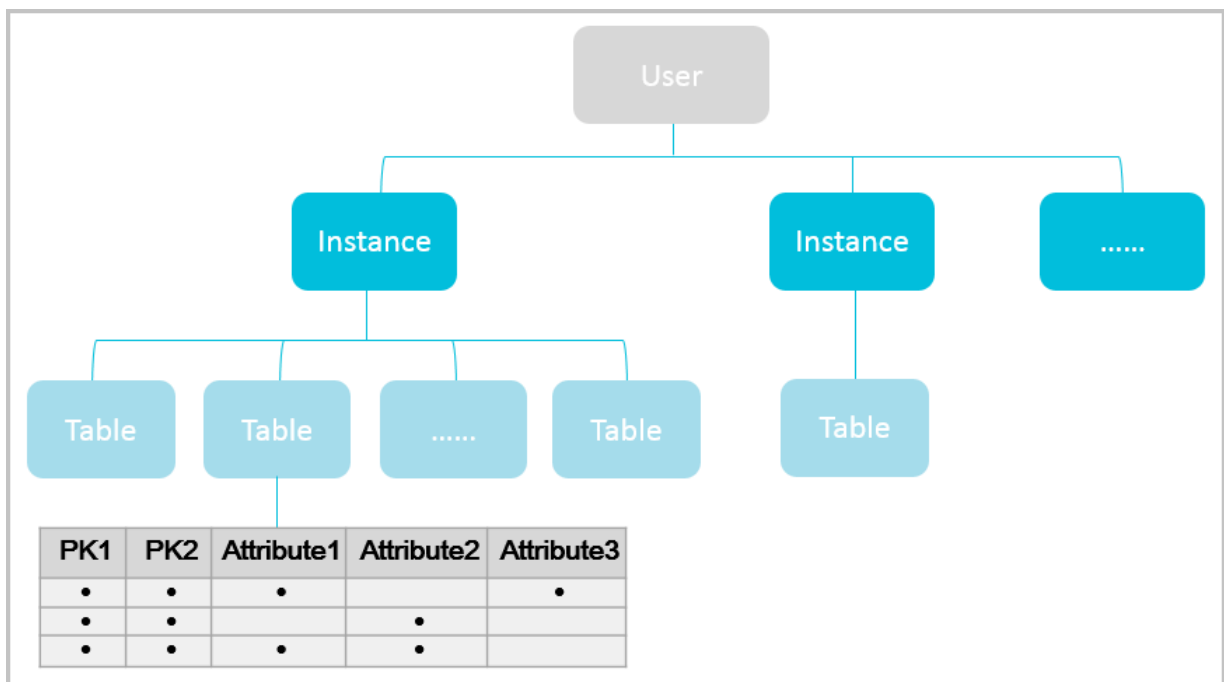
- **Table Worker** is the data engine layer that processes structured data. It uses a primary key to search for or store data. Table Worker supports large-scale access request bursts.
- The bottom layer is the storage layer. Apsara Distributed File System is deployed at this layer. Metadata is stored in Master server roles. The distributed message consistency protocol Paxos is adopted between Master service roles to ensure metadata consistency. In this scenario, efficient distributed file storage and access are achieved. This method guarantees three copies of data stored in the system and system recovery from any hardware or software faults.

## 5.4 Features

### 5.4.1 Users and instances

The following figure shows the Table Store architecture in relation to a user and instances.

Figure 5-1: User and instance architecture



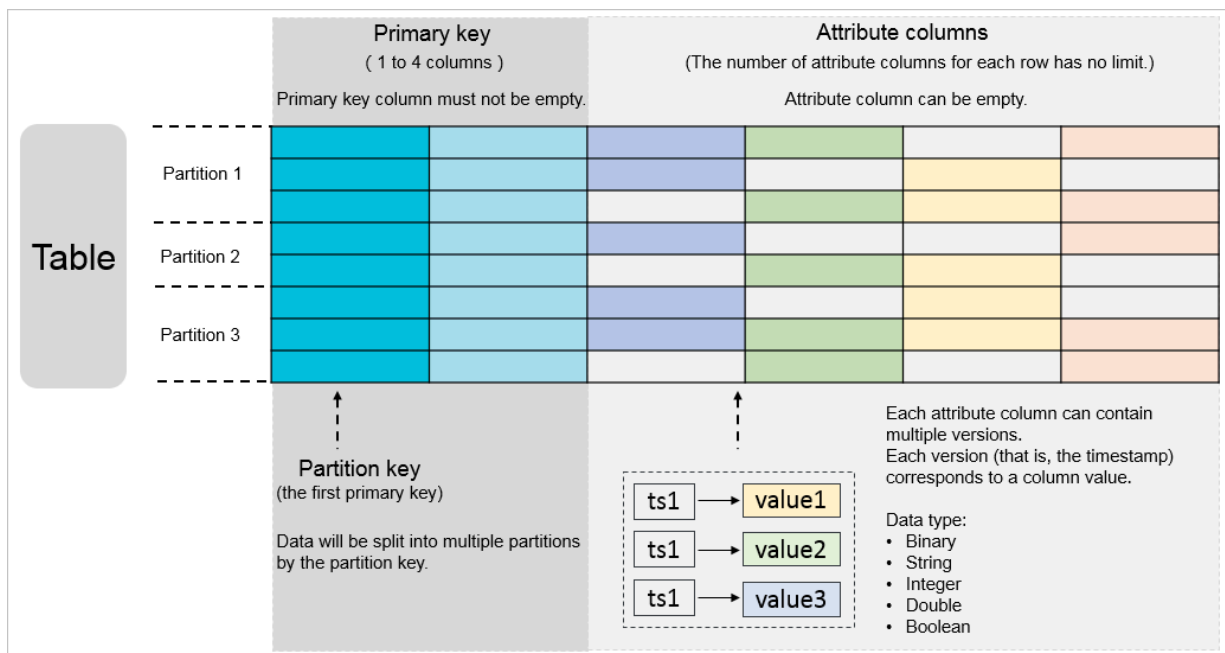
- Users can log on with an Apsara Stack tenant account.
- User operations can be audited in fine granularity.

- Users organize resources through instances. A user can create multiple instances and use each instance to create and manage multiple data tables.
- An instance is the basic unit of multi-tenant isolation.
- User permissions can vary with their roles.

## 5.4.2 Data tables

The following figure shows the data table structure.

Figure 5-2: Data table structure



- A data table is the basic unit of resource allocation.
- A table is a set of rows. A row consists of primary key columns and attribute columns.
- A table partitions data based on the size of the first primary key column.
- All rows in a table must have the same quantity of primary key columns with the same names.
- The quantity of attribute columns in a row is variable. So are the names and data types of the attribute columns.
- There is no limit to the number of attribute columns contained in a row. However, the maximum number of attribute columns where each request can write data to is 1,024.
- A table can contain over hundreds of billions of rows of data.
- A table can store PBs of data.

### 5.4.3 Data partitioning

- A table partitions data based on the size of the first primary key column.
- The rows whose first primary key column values are within the same partition range are allocated to the same partition.
- To improve load balancing, Table Store splits and merges partitions based on specific rules.
- We recommend that you do not store more than 10 GB of data in rows that share the same partition key.

### 5.4.4 Common commands and functions

#### Commands

- **ListTable:** lists all tables in an instance.
- **CreateTable:** creates a table.
- **DeleteTable:** deletes a table.
- **DescribeTable:** obtains attributes of a table.
- **UpdateTable:** updates the reserved read/write throughput configuration of a table.
- **ComputeSplitPointsBySize:** logically partitions all table data into several partitions of the specified size; returns the split points between these partitions and the prompt of the hosts where partitions reside.

#### Functions

- **GetRow:** reads a row of data.
- **PutRow:** inserts a row of data.
- **UpdateRow:** updates a row of data.
- **DeleteRow:** deletes a row of data.
- **BatchGetRow:** reads multiple rows in one or more tables simultaneously.
- **BatchWriteRow:** inserts, updates, or deletes multiple rows in one or more tables simultaneously.
- **GetRange:** reads data from a table within a range.

## 5.4.5 Authorization and access control

### Table Store permissions

**Table Store integrates RAM and VPC to support the following access control mechanisms:**

- **Table-level authorization**
- **API-level access control**
- **Authentication of IP address limits, HTTPS, multi-factor authentication (MFA), and access time limits**
- **Temporary access authorization of STS**
- **VPC access control**

### Apsara Stack Management Console-based permissions

- **Account logons and authentication through Apsara Stack Management Console**
- **Instance creation, management, and deletion functions through GUI**
- **Table creation, management, deletion, and reserved read/write throughput adjustment functions through GUI**
- **Display of table-level monitoring information**

## 6 Network Attached Storage (NAS)

---

### 6.1 What is NAS?

#### 6.1.1 Overview

**Alibaba Cloud NAS provides file storage services to compute nodes, such as ECS instances and Container Service nodes.**

**Alibaba Cloud NAS is a distributed file system with various features, such as shared access, elastic scalability, high reliability, and high performance. Based on POSIX-compliant file APIs, NAS provides a variety of benefits, such as compatibility with different operating systems, shared access, data consistency, and exclusive locks.**

**NAS provides simple and scalable file storage for use with ECS instances. Multiple ECS instances can access a NAS file system at the same time, providing a common data source for workloads and applications running on these instances. With NAS, storage capacity is elastic, and automatically grows or shrinks as you add or remove files.**

#### 6.1.2 Benefits

- **Parallel shared access**

**A file system can be mounted on a maximum of 10 thousand clients at the same time through the NFSv3 or NFSv4 protocol to share data from the same data source.**

- **High performance**

**The throughput of a single file system increases with the growing storage capacity. Without upfront investments in high-end NAS storage devices, NAS helps you to reduce a large number of costs.**

- **Elastic scalability**

**The storage capacity of a NAS file system scales with increasing or decreasing business data. The maximum storage capacity of a file system is 10 PB. Each file system can store a maximum of 1 billion files, and the maximum size of a single file is 32 TB.**

- **High reliability**

**Based on the three-member replica set architecture of Apsara Distributed File System, NAS provides high data reliability to ensure user data security.**

- **Security**

**NAS provides multiple security features, such as VPCs, security groups, access control lists (ACL), and resource access management (RAM) users. This ensures that user data is isolated.**

- **Global namespaces**

**Data of a file system is stored on distributed nodes across the entire NAS cluster, providing a unique namespace.**

### 6.1.3 Scenarios

Scenario 1: shared storage and high availability for SLB

**Your SLB instance is connected to multiple ECS instances. You can store the data of the applications on these ECS instances on a shared NAS instance. This implements data sharing and ensures high availability of the SLB servers.**

Scenario 2: file sharing within an enterprise

**The employees of an enterprise need to access the same datasets for work purposes. The administrator can create a NAS instance and configure different file or directory permissions for users or user groups.**

Scenario 3: data backup

**You want to back up the data stored in the data center to the cloud and use a standard interface to access the cloud storage service. You can back up the data in the data center to a NAS instance.**

Scenario 4: server log sharing

**You want to store the application server logs of multiple compute nodes on the shared file storage. You can store these server logs on a NAS instance for centralized log processing and analysis.**

## 6.2 Technical advantages

**The technical advantages of NAS are reflected in shared access and security.**

### Shared access

**NAS supports the standard NFS and SMB protocols and mainstream operating systems, such as Linux and Windows. You can mount NAS file systems on these operating systems.**

**Multiple compute instances can share access to the same data source with guaranteed strong data consistency.**

### Security

**Provides encryption in transit to ensure data security during transmission.**

**Only allows access to a file system from dedicated networks to ensure maximum access security, such as VPCs and VPNs.**

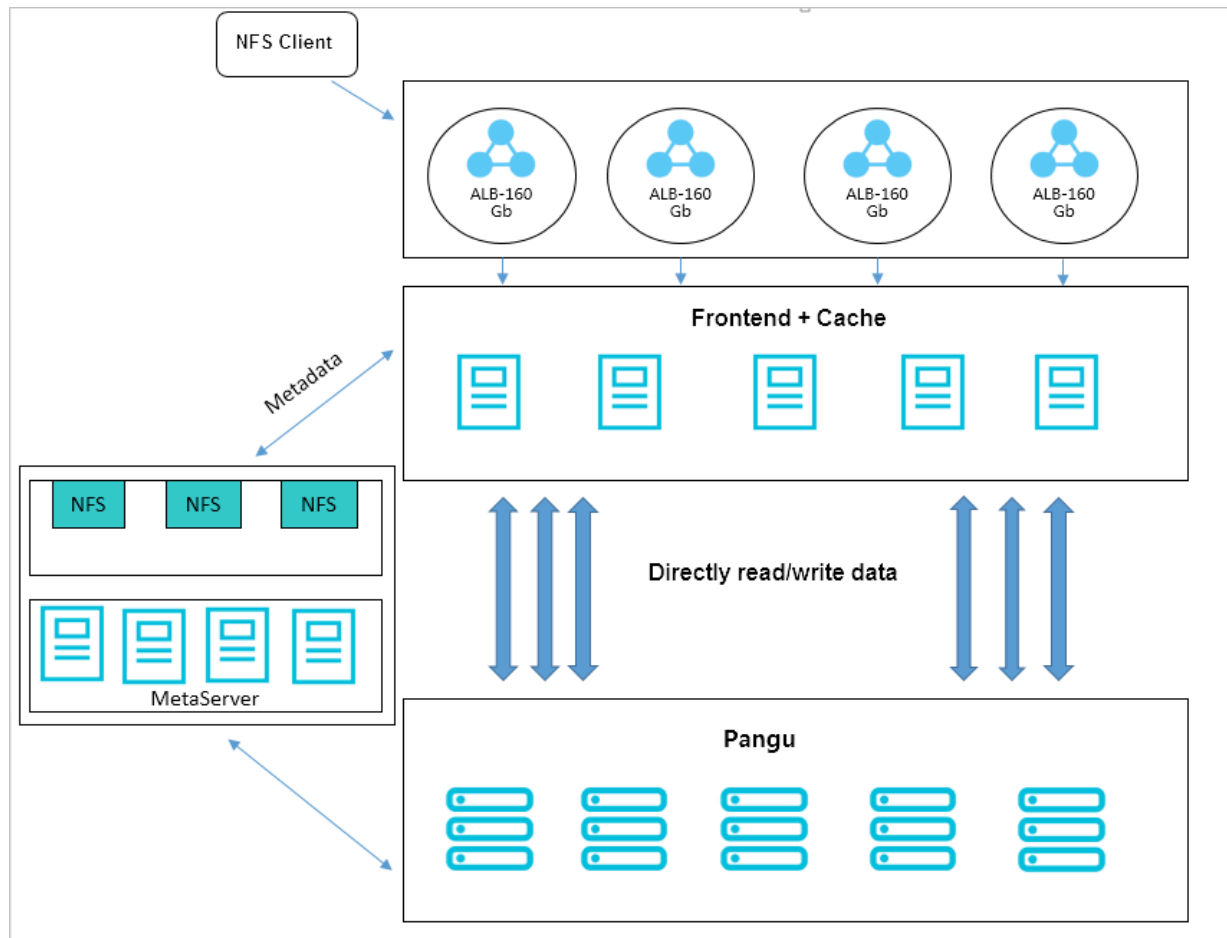
**NAS saves multiple data copies and provides flexible and various backup policies to ensure data security.**

## 6.3 Architecture

**NAS infrastructure is based on Apsara Distributed File System. With this structure, three copies of each piece of file system data are stored on multiple distributed storage nodes. Frontend nodes accept connection requests from NFS clients and cache these requests. As frontend nodes are stateless and distributed, this mechanism ensures high availability of these frontend nodes. The metadata of a NAS file system is stored on a MetaServer. When frontend nodes retrieve metadata from a MetaServer through I/O requests, read and write operations on user data are all performed on backend nodes of Apsara Distributed File System.**

**This structure allows for exclusive and elastic scaling of frontend and backend storage nodes. On the premise of ensuring high availability, the structure enables NAS to achieve high concurrency and low latency throughput.**

Figure 6-1: System architecture



## 6.4 Features and principles

### 6.4.1 Feature overview

NAS supports the NFSv3 and NFSv4 protocols. You can use NAS without making any changes to your existing applications. You can use either protocol to access NAS instances for the following purposes: business file sharing, backend file storage for office automation systems, enterprise database backup and storage, business



system log storage and analysis, website data storage and distribution, and data storage during business system development and testing.

Figure 6-2: Features



## 6.4.2 Features

### Seamless integration

**Network Attached Storage (NAS) supports the NFSv3 and NFSv4 protocols and provides access through standard file system interfaces. Most applications and workloads can seamlessly work with NAS without any change.**

### Shared access

**A NAS file system can be accessed by multiple compute nodes. NAS supports simultaneous access from multiple compute nodes. Therefore, a NAS file system**

**is well suited if your application is deployed across multiple ECS instances or Container Service nodes.**

#### Access control

**NAS provides multiple security control mechanisms to ensure data security of its file systems. These mechanisms include but are not limited to: network isolation by VPCs, user isolation in classic networks, standard permission control for file systems, security group based access control, and RAM user authorization.**

#### Linear performance

**NAS allows your applications to achieve optimal storage performance of high throughput and IOPS with consistent low latency. The storage performance linearly improves as the storage capacity increases. This meets the high requirements imposed by business growth on both storage capacity and storage performance.**

### 6.4.3 Terms

#### mount point

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

#### permission group

**The permission group is a whitelist mechanism provided by NAS. You can add rules to a permission group of a NAS instance to allow users from specified IP addresses or address segments to access the NAS instance with different permissions.**



#### **Note:**

**Each mount point must be bound with a permission group.**

#### authorized object

**An authorized object is an attribute of the permission group rule. It specifies the IP address or address segment to which the permission group rule is applied. In a VPC, an authorized object can be a single IP address or an address segment. In a classic network, an authorized object must be an IP address, generally the intranet IP address of an ECS instance.**

## 7 Apsara File Storage for HDFS

---

### 7.1 What is Apsara File Storage for HDFS?

#### 7.1.1 Terms

**Apsara File Storage for HDFS is a file storage service for computing resources such as Alibaba Cloud ECS instances and Container Service. Apsara File Storage for HDFS provides data management and access solutions similar to the Hadoop Distributed File System (HDFS). You can use Apsara File Storage for HDFS without modifying existing big data applications. Apsara File Storage for HDFS offers various features such as unlimited capacity, performance expansion, single namespace, multi-party sharing, high reliability, and high availability.**

**After creating an Apsara File Storage for HDFS instance, you can access the file system through standard HDFS protocol interfaces in computing resources, such as ECS and Container Service instances. In addition, multiple compute nodes can simultaneously access the same Apsara File Storage for HDFS to share files and directories.**

#### 7.1.2 Benefits

Comparison with a traditional HDFS

**A traditional HDFS is designed to allow high-throughput data access to support applications on large-scale datasets. Apsara File Storage for HDFS has advantages over the traditional HDFS in terms of system elasticity, small file storage performance, and multi-tenant support.**

- System elasticity

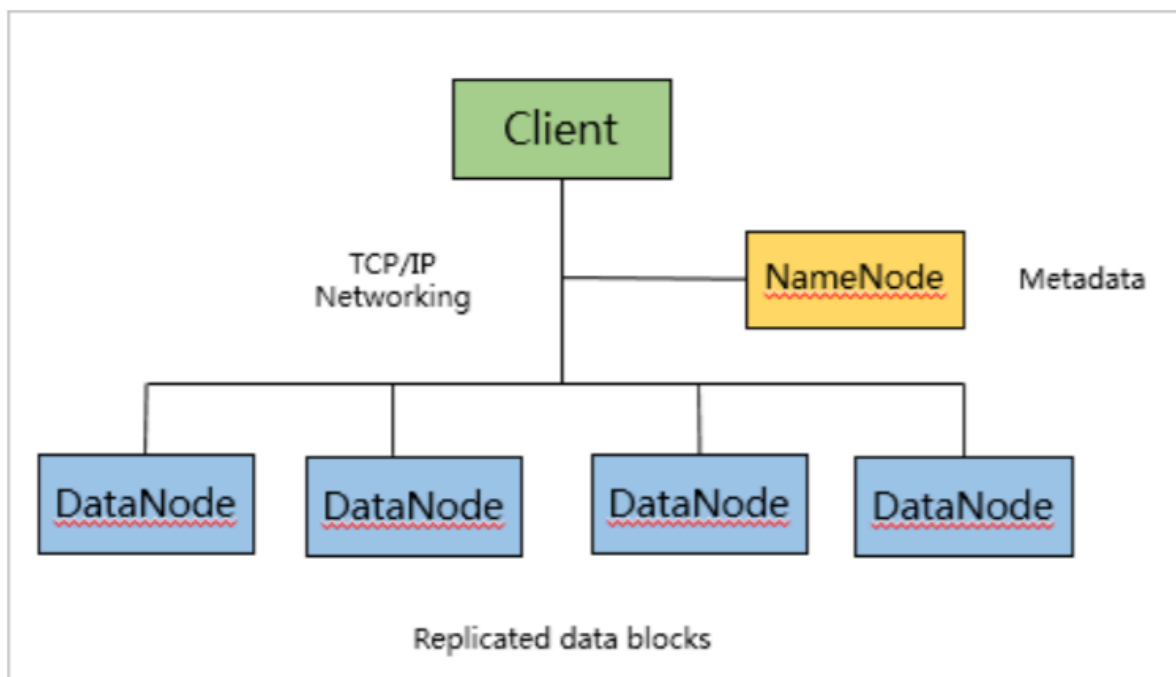
A DataNode of a traditional HDFS must provide both storage and computing capabilities. Therefore, it is not flexible enough for planning and capacity scaling.

The following figure shows the deployment architecture of Apsara File Storage for HDFS.

As shown in the figure, Apsara File Storage for HDFS is deployed in three clusters : the storage cluster, the compute cluster, and the common service cluster. The storage cluster and the compute cluster are independent of each other, and can be planned and scaled separately. After computing operations are complete, resources can be released while stored data can be retained. This way, the system elasticity is improved for Apsara File Storage for HDFS.

- Small file storage performance

The following figure shows the architecture of a traditional HDFS.



As shown in the figure, a file in HDFS is split into one or more blocks and these blocks are stored in a set of DataNodes. Metadata of file blocks are maintained in a single NameNode. For all file I/O requests, clients must obtain metadata from the NameNode and then write data to each DataNode in chain mode. HDFS was

designed to be used with Hard Disk Drives (HDDs) and cannot take advantage of flash media, such as Solid State Drive (SSD) and NVMe Express (NVMe).

The architecture of HDFS allows the system to stream and read large datasets. However, HDFS is not suited for processing small files because large amounts of small files generate large amounts of metadata, which can exhaust the resources of the NameNode. In addition, all file operation requests must be sent through the NameNode, which lengthens the latency of data access.

The architecture of Apsara File Storage for HDFS caters to the demand for storage of small files. The system significantly improves the throughput of small files and storage efficiency through support for hierarchical storage, optimization of metadata, and optimization of write streams.

- **Multi-tenancy**

The traditional HDFS is designed for the single-cluster or single-tenant usage in on-premises computing environments. DataNodes provide storage and computing capabilities for local data. Therefore, the computing system is closely bound to the storage system.

Compared with the traditional HDFS, Apsara File Storage for HDFS is based on a cloud computing environment with native support for cloud computing. As a cloud storage system, Apsara File Storage for HDFS establishes multiple file system instances for multiple tenants. Each instance supports operations through multiple compute clusters.

### 7.1.3 Scenarios

This topic describes the usage scenarios of Apsara File Storage for HDFS.

Scenario 1: shared storage and high availability

We recommend that you use Apsara File Storage for HDFS to store files if you have the following business requirements:

- You need to access files in shared mode.
- You demand high availability of files.

Apsara File Storage for HDFS supports standard HDFS protocols. You can use standard HDFS interfaces to store files in real time or in batches to Apsara File Storage for HDFS.

## Scenario 2: big data analytics and machine learning

**In big data analytics and machine learning scenarios, applications require high throughput performance and short latency for data access. Apsara File Storage for HDFS supports high-throughput and low-latency access. You do not need to migrate data to local computing resources. Therefore, Apsara File Storage for HDFS is recommended in this scenario.**

**After data is stored in Apsara File Storage for HDFS, ECS instances or other computing resources can directly access the data. You can deploy Hadoop or other machine learning applications on multiple computing resources so that applications can access data directly through the HDFS interfaces to perform online or offline computation. You can also export the calculation results to an Apsara File Storage for HDFS instance and store them permanently.**

## 7.2 Design philosophy

**This topic introduces the background and design philosophy of Apsara File Storage for HDFS.**

### Background

**With the application of big data and the development of Hadoop technology, there is an increasing demand from users for distributed file systems. Compared with the traditional HDFS, an ideal file system must have the following features:**

- **Cloud data intercommunication**
- **Calculation of stored data at any time**
- **Disaster recovery**
- **High performance and low cost**

**To address these needs, Alibaba developed high-performance Apsara File Storage for HDFS that is compatible with Hadoop.**

### Design philosophy

**Apsara File Storage for HDFS is designed based on the following concepts:**

- **Apsara File Storage for HDFS is compatible with the Hadoop ecosystem. Hadoop applications can be seamlessly connected to large amounts of data on the cloud.**
- **With the integrated design of software and hardware, the system features extreme end-to-end performance and low cost.**

- **The system can interface with Apsara Stack storage services such as Object Storage Service (OSS) and Apsara File Storage (NAS) so that its data can be accessed from ECS and MaxCompute at any time.**
- **The system provides users with an excellent experience due to its smart management and O&M capabilities.**

#### Features

**Apsara File Storage for HDFS has the following features:**

- **Compatibility with Hadoop File System APIs**
- **High availability based on three-zone disaster recovery**
- **Linear scalability**
- **Automated O&M and management**

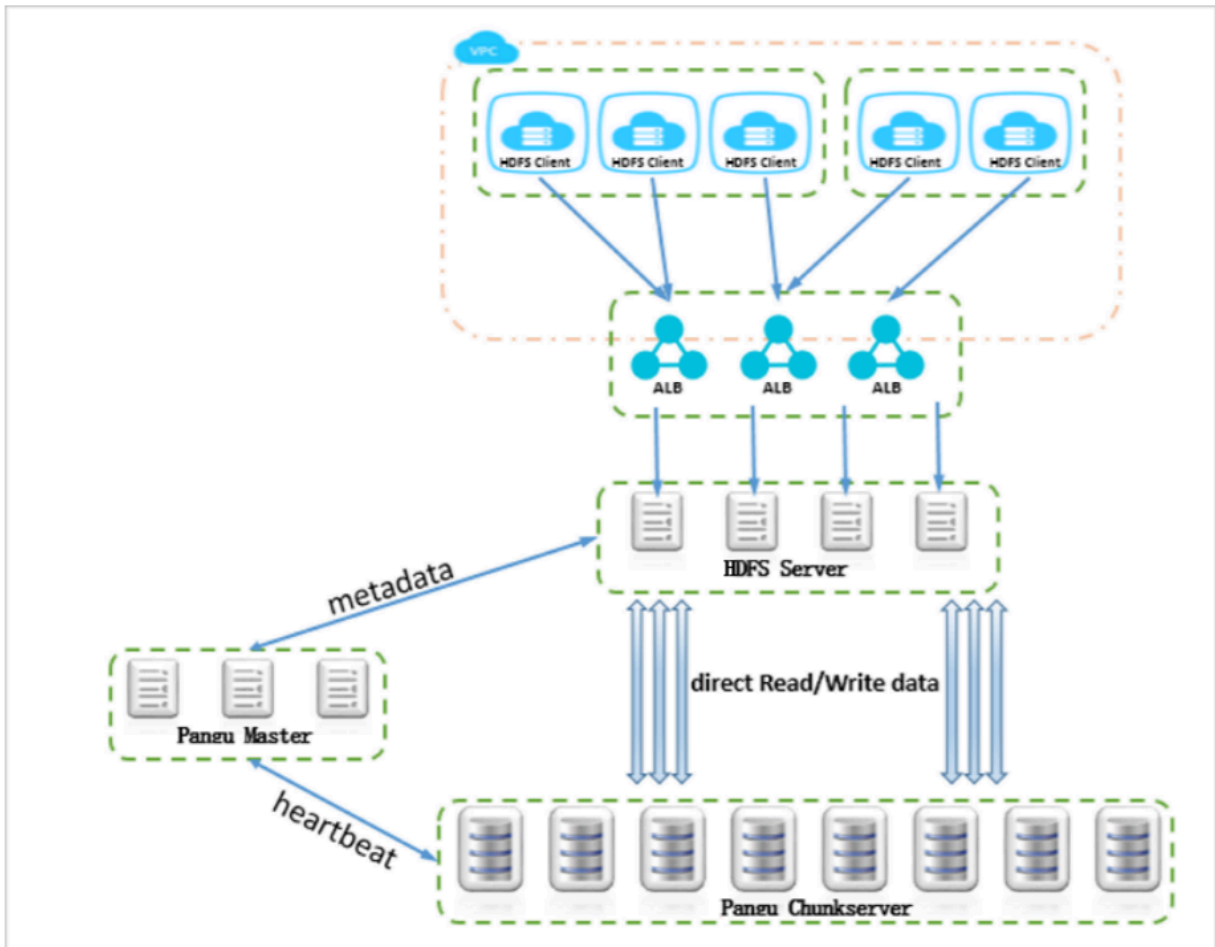
### 7.3 Architecture

**This topic introduces the system architecture of Apsara File Storage for HDFS.**

**The architecture of Apsara File Storage for HDFS is divided into two parts: frontend and backend.**

**The backend is based on Apsara Distributed File System. Data is stored in Apsara Distributed File System as multiple copies. The frontend access nodes of Apsara File Storage for HDFS receive connection requests from ECS (for example, Hadoop computing applications such as MapReduce and Spark) or Container Service instances, and cache data. Apsara Distributed File System also manages metadata and data of Apsara File Storage for HDFS.**

**The overall architecture of Apsara File Storage for HDFS is as follows:**



## 7.4 Benefits

This topic introduces the benefits of Apsara File Storage for HDFS in terms of access, performance, and cost.

### Easy access

- **Apsara File Storage for HDFS is compatible with Hadoop FileSystem APIs. It can be accessed from existing Hadoop applications at no additional cost.**
- **Data from Apsara File Storage for HDFS can be pooled with data from Apsara Stack storage services.**
- **Apsara File Storage for HDFS can be conveniently accessed from MaxCompute.**

### High performance

- **Apsara File Storage for HDFS is optimized for small files.**
- **Apsara File Storage for HDFS makes full use of the hardware advantages of new-generation flash media.**



#### Cost reduction

- **Apsara File Storage for HDFS supports cloud-based storage and automatic scaling, which helps reduce storage costs.**
- **Data from Apsara File Storage for HDFS can be pooled with data from storage services to immediately enable computing capabilities, which saves time.**
- **Large amounts of small files can be stored, reducing data management costs.**

## 8 ApsaraDB for RDS

---

### 8.1 What is ApsaraDB for RDS?

ApsaraDB for RDS is a stable, reliable, and automatically scaling online database service.

Based on the distributed file system and high-performance storage, ApsaraDB for RDS allows you to easily perform database operations and maintenance with its complete set of solutions for disaster recovery, backup, restoration, monitoring, and migration.

ApsaraDB for RDS supports three storage engines: MySQL, PostgreSQL, and PPAS . These storage engines can help you create database instances suitable to your business needs.

#### ApsaraDB RDS for MySQL

Originally based on a branch of MySQL, ApsaraDB RDS for MySQL has proven its performance and throughput during the high-volume concurrent traffic of Double 11. ApsaraDB RDS for MySQL provides whitelist configuration, backup and restoration, transparent data encryption, data migration, and management for instances, accounts, and databases. It also provides the following advanced features:

- **Read-only instance:** In scenarios where RDS has a small number of write requests but a large number of read requests, you can enable read/write splitting to distribute read requests away from the primary instance. Read-only instances allow ApsaraDB RDS for MySQL 5.6 to automatically scale the reading capability and increase the application throughput when a large amount of data is being read.
- **Read/write splitting:** The read/write splitting feature provides an extra read/write splitting endpoint. This endpoint enables an automatic link for the primary instance and all its read-only instances. An application can use this method to read and write data by connecting to the read/write splitting endpoint. Write requests are automatically distributed to the primary instance while read requests are distributed to read-only instances based on their

weights. To scale up the reading capacity of the system, you can add more read-only instances.

- **Data compression:** ApsaraDB RDS for MySQL 5.6 allows you to compress data by using the TokuDB storage engine. Data transferred from the InnoDB storage engine to the TokuDB storage engine can be reduced by 80% to 90% in volume. 2 TB of data in InnoDB can be compressed to 400 GB or less in TokuDB. In addition to data compression, TokuDB supports transaction and online DDL operations. TokuDB is compatible with MyISAM and InnoDB applications.

#### ApsaraDB RDS for PostgreSQL

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

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

#### ApsaraDB RDS for PPAS

Postgres Plus Advanced Server (PPAS) is a stable, secure, and scalable enterprise-class relational database. Based on PostgreSQL, PPAS features enhanced performance, application solutions, and compatibility. It is able to directly run Oracle applications. You can run enterprise-class applications on PPAS in a stable and cost-effective manner.

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

## 8.2 Benefits

**This topic describes the benefits of ApsaraDB for RDS.**

### High availability

**The high-availability service uses the Detection, Repair, and Notice modules to ensure the availability of data link services. It also processes internal database exceptions. The high-availability service calls the proxy API operation to re-attach RDS instances based on the heartbeat monitoring status of the primary and secondary databases.**

### Backup service

**The backup service provides offline data backup, dumping, and recovery. It uploads binlogs of RDS instances to OSS buckets for instance backup.**

### Monitoring service

**The monitoring service uses the agents deployed on the hosts to monitor the status of the host cluster and instances. It provides various monitoring services at the physical layer, network layer, and application layer to ensure service availability.**

### Scheduling service

**The scheduling service allocates resources and manages the instance version. The scheduling service balances the resource usage of instances on each host, and allocates and integrates underlying RDS resources when instances are activated and migrated. When you use the RDS console or an API operation to create an instance, the scheduling service calculates the most suitable host to carry traffic to and from the instance. After you have created, deleted, and migrated instances multiple times, the scheduling service calculates the resource fragmentation rate in a zone and periodically integrates resources to improve service capabilities of the zone.**

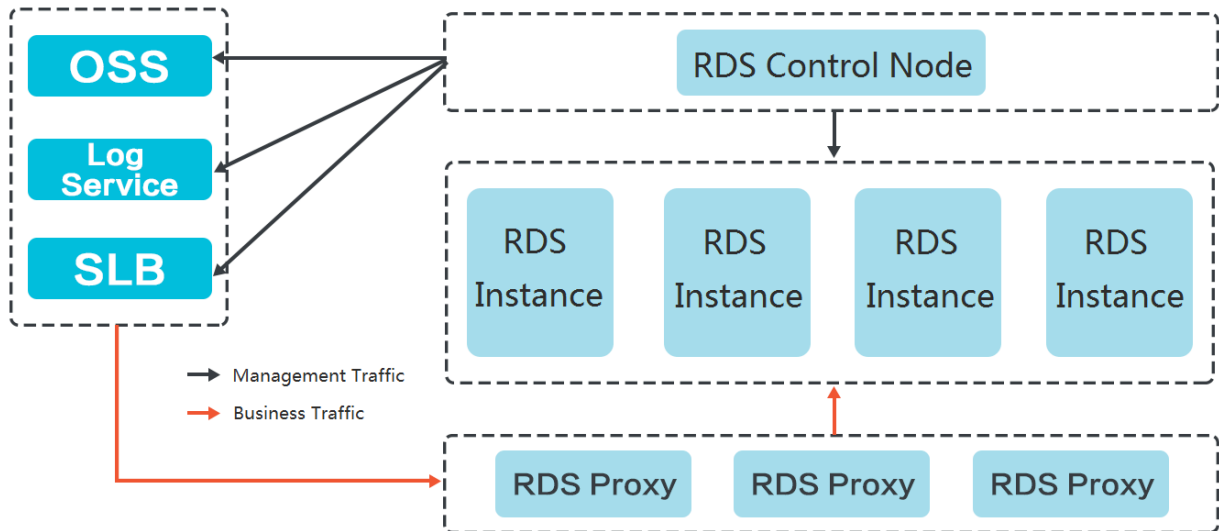
### Read/write splitting

**In the middle of the RDS data link, the RDS Proxy receives traffic from the client by using the SLB VIP and connects to databases. The RDS Proxy provides semantic support and distributes sessions to different instances to implement read/write splitting.**

### 8.3 Architecture

The following figure shows the system architecture of ApsaraDB for RDS.

Figure 8-1: RDS system architecture



Basic components of RDS include the RDS control node, RDS instance, and RDS Proxy. They are described as follows:

- **RDS control node:** serves as the RDS system controller. It controls the entire RDS cluster, including database resources, monitoring, O&M tasks, backup, and HA.
- **RDS instance:** serves as the database service node.
- **RDS Proxy:** provides a number of features such as data routing, traffic detection, and session persistence.

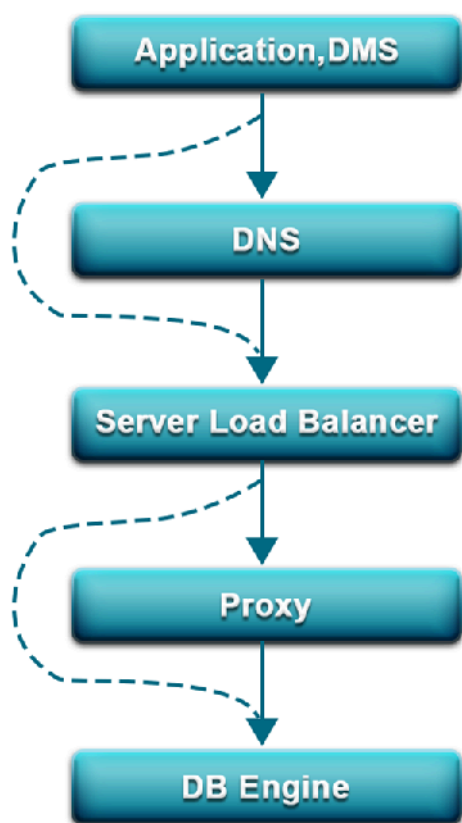
Business traffic is directed to RDS instances through the RDS Proxy modules based on the SLB VIP and port number mapped to the domain name as well as the logon information of the instances.

## 8.4 Features and principles

### 8.4.1 Data link service

The data link service allows you to add, delete, modify, and query the table schema and data.

Figure 8-2: RDS data link service



#### DNS

The DNS module can dynamically resolve domain names to IP addresses. Therefore, IP address changes do not affect the performance of RDS instances.

For example, the domain name of an ApsaraDB for RDS instance is `test.rds.aliyun.com`, and its corresponding IP address is `10.1.1.1`. The instance can be accessed when either `test.rds.aliyun.com` or `10.1.1.1` is configured in the connection pool of a program.

After a zone migration or version upgrade is performed for this ApsaraDB for RDS instance, the IP address may change to `10.1.1.2`. 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 10.1.1.1 is configured in the connection pool, the instance will no longer be accessible.

## SLB

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

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

## Proxy

The Proxy module provides a number of features including data routing, traffic detection, and session persistence.

- **Data routing:** aggregates the distributed complex queries found in big data scenarios and provides the corresponding capacity management capabilities.
- **Traffic detection:** reduces SQL injection risks and supports SQL log backtracking when necessary.
- **Session persistence:** prevents interruptions to the database connection when faults occur.

## DB Engine

The following table describes the mainstream database protocols supported by RDS.

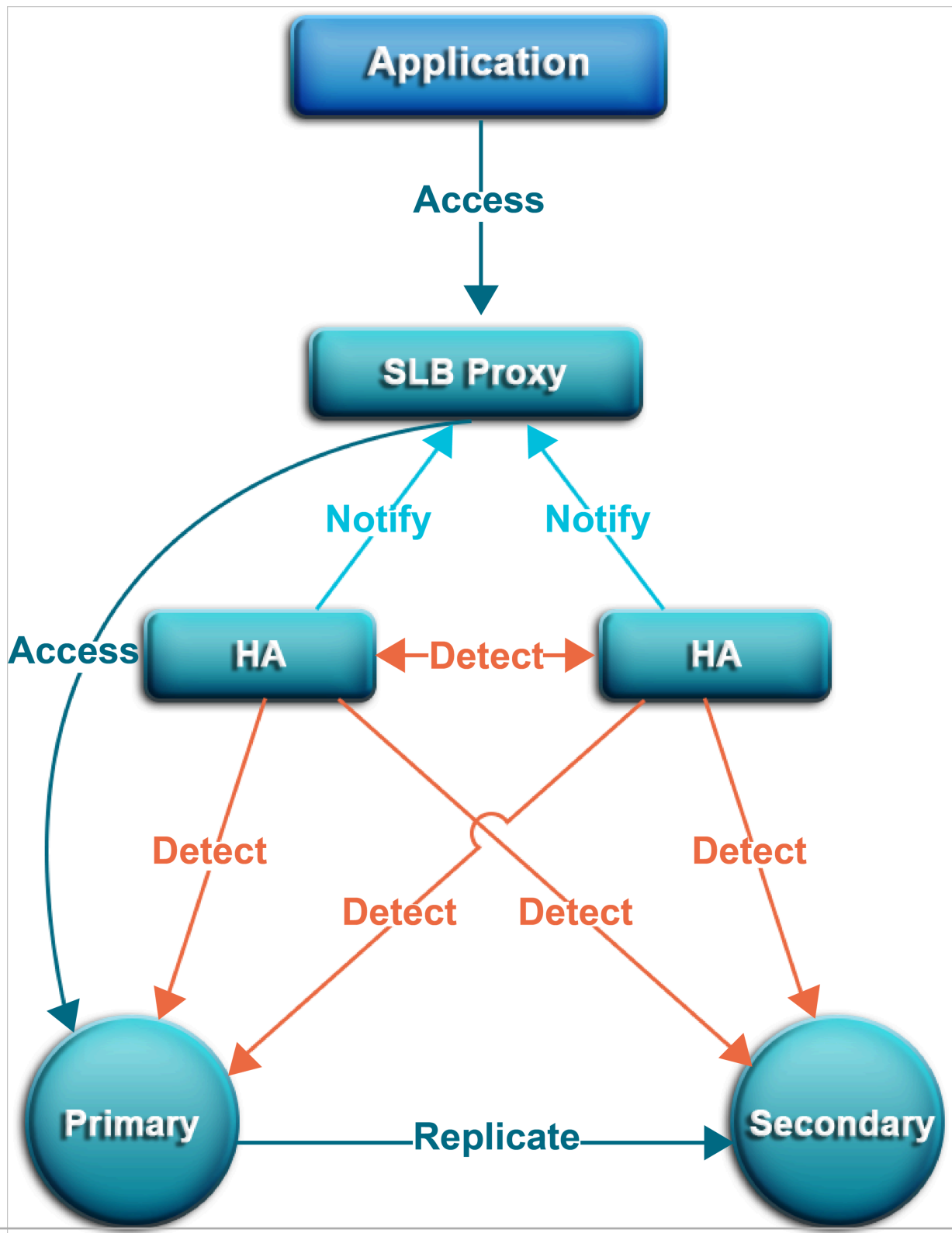
Table 8-1: RDS database protocols

RDBMS	Version
MySQL	5.6 or 5.7 (including read-only instances)
PostgreSQL	9.4
PPAS	9.3 or 9.6

### 8.4.2 High-availability service

The high-availability (HA) service ensures the availability of data link services and processes internal database exceptions. The HA service is implemented by multiple HA nodes.

Figure 8-3: RDS HA service





## 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 every 8 to 10 seconds 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 and helps avoid false positives caused by exceptions such as network jitter. Failover can be completed within 30 seconds.**

## 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 either node during normal operations such as in the following scenarios:**

- **Automatically restores primary/secondary replication after a disconnection.**
- **Automatically repairs table-level damage to the primary or secondary node.**
- **Automatically saves and repairs the primary or secondary node in case of crashes.**

## Notice

**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, as shown in the following example.**

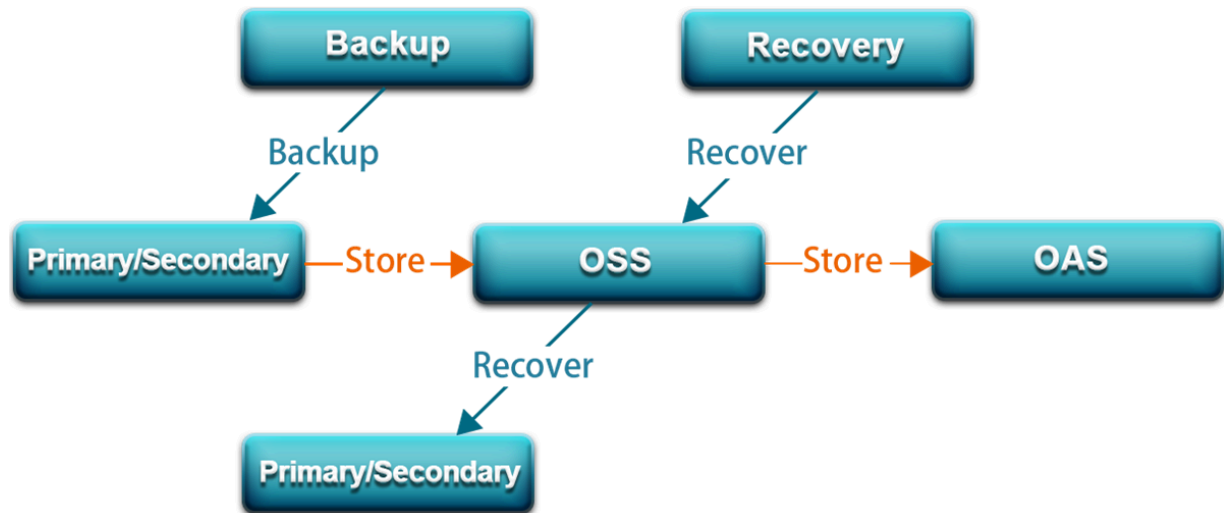
**Assume that the Detection module has discovered a problem with the primary node and has instructed the Repair module to resolve the problem. If the Repair module fails to resolve the problem, it instructs the Notice module to redirect traffic. The Notice module forwards the switching request to the SLB or Proxy module. Traffic is then redirected to the secondary node.**

**Meanwhile, the Repair module creates a new secondary node on a different host and synchronizes the change back to the Detection module. The Detection module rechecks the health status of the instance to ensure it is healthy.**

### 8.4.3 Backup service

This service supports offline data backup, dumping, and recovery.

Figure 8-4: RDS backup service



#### Backup

The Backup module compresses and uploads data and logs on both the primary and secondary nodes. ApsaraDB for RDS uploads backup files to OSS and dumps the backup files to a more cost-effective and persistent Archive Storage system. When the secondary node is operating normally, backup is always created on the secondary node so as not to affect the services on the primary node. When the secondary node is unavailable or damaged, the Backup module creates backups on the primary node.

#### Recovery

The Recovery module restores backup files stored in OSS to a destination node. The Recovery module provides the following features:

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

#### Storage

The Storage module uploads, dumps, and downloads backup files.

All backup data is uploaded to OSS for storage. You can obtain temporary links to download backups as necessary.



**Note:**

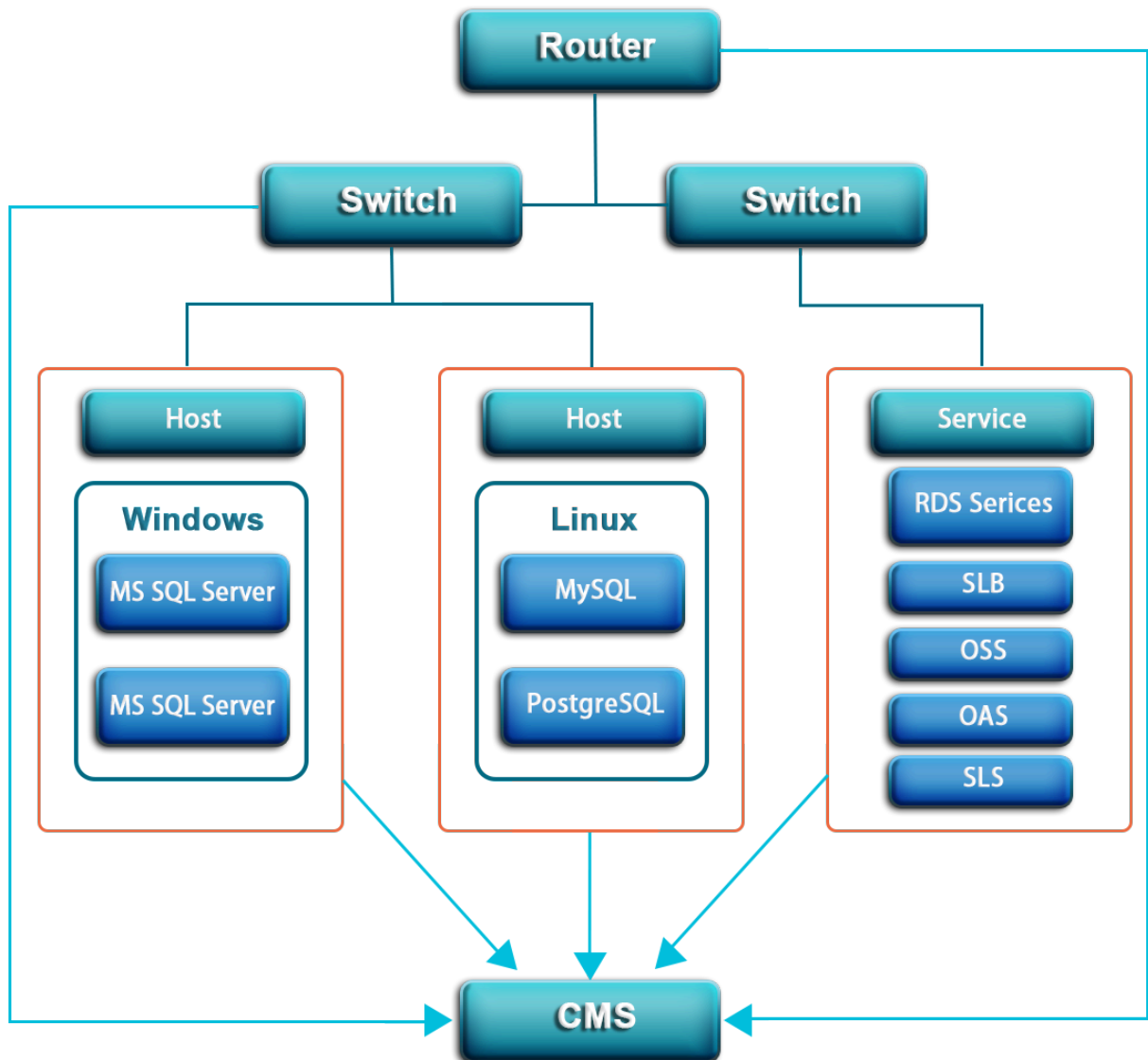
ApsaraDB RDS for PPAS does not allow you to download backup files.

In certain scenarios, the Storage module allows you to dump backup files from OSS to Archive Storage for more cost-effective and longer-term offline storage.

#### 8.4.4 Monitoring service

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

Figure 8-5: RDS monitoring service



## Service

**The Service module tracks the status of services that RDS depends on, such as Server Load Balancing (SLB), OSS, Archive Storage, and log service, to ensure they are operating properly. Monitored metrics include functionality and response time. The Service module also uses logs to determine whether internal ApsaraDB for RDS services are operating properly.**

## Network

**The Network module tracks statuses at the network layer. The monitored items include:**

- **The connectivity between ECS and ApsaraDB for RDS**
- **The connectivity between physical RDS servers**
- **The rates of packet loss on the VRouter and VSwitch**

## OS

**The OS module tracks the statuses of hardware and the OS kernel. The monitored items include:**

- **Hardware maintenance:** The OS module constantly checks the operating status of the CPU, memory, motherboard, and storage device. It can predict faults in advance 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 for RDS instances:**

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

## 8.4.5 Scheduling service

**The scheduling service allocates resources and manages the instance version.**

### Resource

**The Resource module schedules resources, and allocates and integrates underlying RDS resources when instances are activated or migrated. When you use the RDS console or an API operation to create an instance, the Resource module calculates the most suitable host to carry traffic to and from the instance. A similar process occurs during ApsaraDB for RDS instance migration.**

**After you have created, deleted, and migrated instances multiple times, the scheduling service calculates the resource fragmentation rate and periodically integrates resources to improve service capabilities.**

## 8.4.6 Migration service

**The migration service can migrate data from your on-premises databases to ApsaraDB for RDS.**

### DTS

**DTS supports data transmission between relational databases, NoSQL databases, and big data OLAP databases.**

**DTS is a data exchange service that streamlines data migration, real-time synchronization, and subscription. DTS is dedicated to implementing remote and millisecond-speed asynchronous data transmission in various scenarios. Based on the active geo-redundancy architecture designed for Double 11, DTS can implement security, scalability, and high availability by providing real-time data streams to up to thousands of downstream applications.**

## 9 KVStore for Redis

---

### 9.1 What is KVStore for Redis?

**KVStore for Redis is an online key-value storage service compatible with open-source Redis protocols. KVStore for Redis supports various types of data, such as strings, lists, sets, sorted sets, and hash tables. The service also supports advanced features, such as transactions, message subscription, and message publishing. Based on the hybrid storage of memory and hard disks, KVStore for Redis can provide high-speed data read/write capability and support data persistence.**

**As a cloud computing service, KVStore for Redis works with hardware and data deployed in the cloud, and provides comprehensive infrastructure planning, network security protections, and system maintenance services.**

#### 9.1.1 Scenarios

##### Game industry applications

**KVStore for Redis can be an important part of the business architecture for deploying a game application.**

##### **Scenario 1: KVStore for Redis works as a storage database**

**The architecture for deploying a game application is simple. You can deploy a main program on an ECS instance and all business data on a KVStore for Redis instance. The KVStore for Redis instance works as a persistent storage database. KVStore for Redis supports data persistence, and stores redundant data on primary and secondary nodes.**

##### **Scenario 2: KVStore for Redis works as a cache to accelerate connections to applications**

**KVStore for Redis can work as a cache to accelerate connections to applications. You can store data in a Relational Database Service (RDS) database that works as a backend database.**

**Reliability of the KVStore for Redis service is vital to your business. If the KVStore for Redis service is unavailable, the backend database is overloaded when processing connections to your application. KVStore for Redis provides a two-node**

hot standby architecture to ensure high availability and reliability of services. The primary node provides services for your business. If this node fails, the system automatically switches services to the secondary node. The complete failover process is transparent.

#### Live video applications

In live video services, KVStore for Redis works as an important measure to store user data and relationship information.

Two-node hot standby ensures high availability

KVStore for Redis uses the two-node hot standby method to maximize service availability.

Cluster editions eliminate the performance bottleneck

KVStore for Redis provides cluster instances to eliminate the performance bottleneck that is caused by Redis single-thread mechanism. Cluster instances can effectively handle traffic bursts during live video streaming and support high-performance requirements.

Easy scaling relieves pressure at peak hours

KVStore for Redis allows you to easily perform scaling. The complete upgrade process is transparent. Therefore, you can easily handle traffic bursts at peak hours

.

#### E-commerce industry applications

In the e-commerce industry, the KVStore for Redis service is widely used in the modules such as commodity display and shopping recommendation.

Scenario 1: rapid online sales promotion systems

During a large-scale rapid online sales promotion, a shopping system is overwhelmed by traffic. A common database cannot properly handle so many read operations.

However, KVStore for Redis supports data persistence, and can work as a database system.

Scenario 2: counter-based inventory management systems

In this scenario, you can store inventory data in an RDS database and save count data to corresponding fields in the database. In 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. Based on solid-state drive (SSD) high-performance storage, the system can provide a high-level data storage capacity.

## 9.2 Benefits

### High performance

- **Supports cluster features and provides cluster instances of 128 GB or higher to meet large capacity and high performance requirements.**
- **Provides primary/secondary instances of 32 GB or smaller to meet general capacity and performance requirements.**

### Elastic scaling

- **Easy scaling of storage capacity: you can scale instance storage capacity in the KVStore for Redis console based on business requirements.**
- **Online scaling without interrupting services: you can scale instance storage capacity on the fly. This does not affect your business.**

### Resource isolation

**Instance-level resource isolation provides enhanced stability for individual services.**

### Data security

- **Persistent data storage: based on the hybrid storage of memory and hard disks, KVStore for Redis can provide high-speed data read/write capability and support data persistence.**
- **Primary/secondary backup and failover: KVStore for Redis backs up data on both a primary node and a secondary node and supports the failover feature to prevent data loss.**
- **Access control: KVStore for Redis requires password authentication to ensure secure and reliable access.**
- **Data transmission encryption: KVStore for Redis supports encryption based on Secure Sockets Layer (SSL) and Secure Transport Layer (TLS) to secure data transmission.**



#### High availability

- **Primary/secondary structure:** each instance runs in this structure to eliminate the possibility of single points of failure (SPOFs) and guarantee high availability.
- **Automatic detection and recovery of hardware faults:** the system automatically detects hardware faults and performs the failover operation within several seconds. This can minimize your business losses caused by unexpected hardware faults.

#### Easy to use

- **Out-of-the-box service:** KVStore for Redis requires no setup or installation. You can use the service immediately after purchase to ensure efficient business deployment.
- **Compatible with open-source Redis:** KVStore for Redis is compatible with Redis commands. You can use any Redis clients to easily connect to KVStore for Redis and perform data operations.

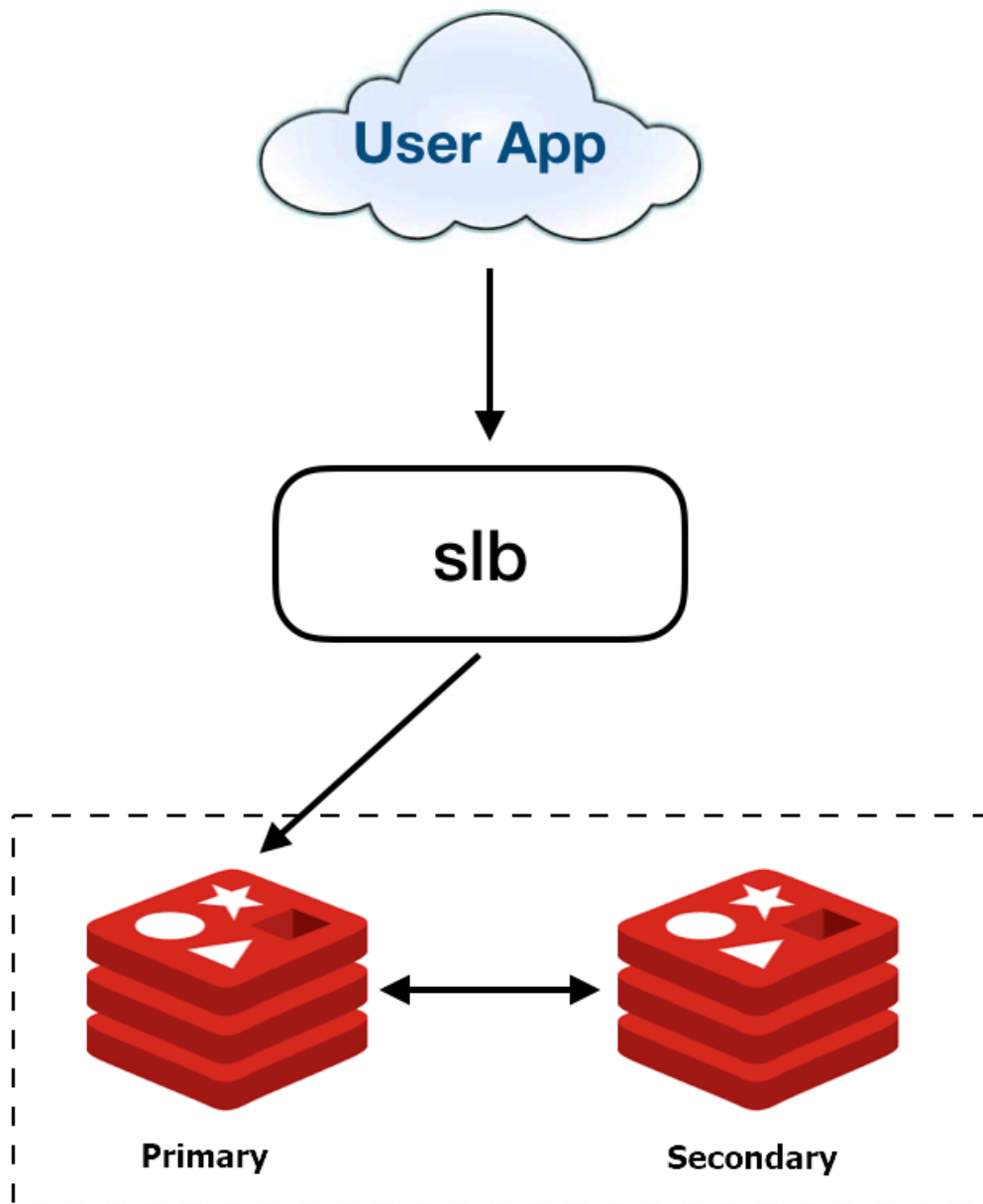
## 9.3 Architectures

### 9.3.1 Overall system architecture

**KVStore for Redis provides primary/secondary and cluster architecture modes.**

#### Primary/secondary architecture

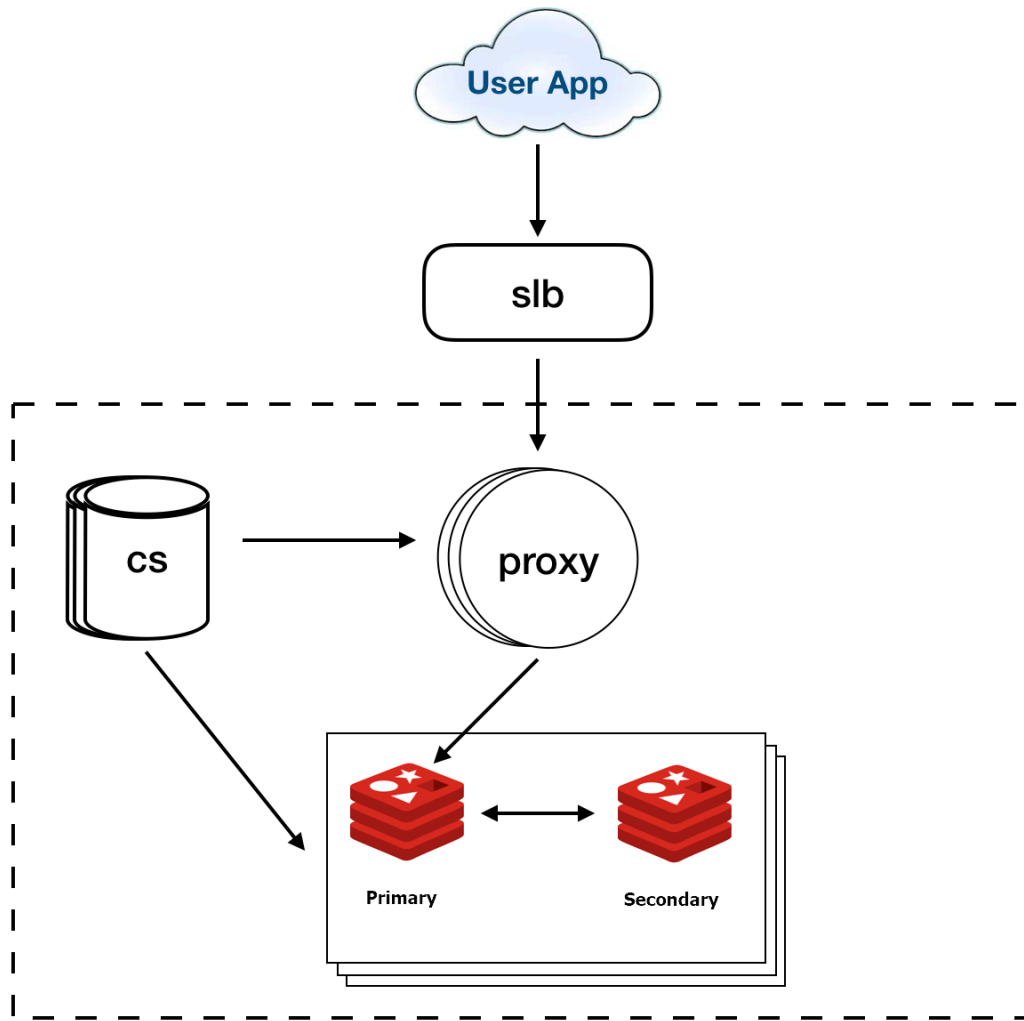
**The following figure shows the primary/secondary architecture.**



**The primary/secondary architecture consists of a primary KVStore for Redis database and a secondary KVStore for Redis database. You can directly access the primary database through an SLB connection.**

Cluster architecture

**The following figure shows the cluster architecture.**



The cluster architecture consists of three components: redis-config (cs), redis-proxy (proxy), and Redis.

The cluster architecture consists of multiple cs nodes, proxy nodes, and primary /secondary Redis nodes. After you access the proxy component through an SLB connection, the proxy component forwards request routes to a shard of the primary Redis database.

### 9.3.2 Components

This topic describes the components of KVStore for Redis and how these components provide services.

#### redis-config

The redis-config (cs) component stores the metadata and topology information of the cluster, and performs cluster operations and maintenance. The cs component

keeps checking heartbeat messages with the Redis and proxy components, and synchronizes metadata and topology information of clusters to redis and proxy.

#### redis-proxy

The redis-proxy (proxy) component is the proxy server that connects your client to a Redis server and that implements Redis protocols. The proxy component can authenticate user identities, forward request routes, provide slow and audit logs, and collect monitoring data at an interval of several seconds.

#### Redis kernel

Alibaba Cloud has optimized the proprietary Redis kernel and developed cloud features based on the open-source kernel from the Redis community.

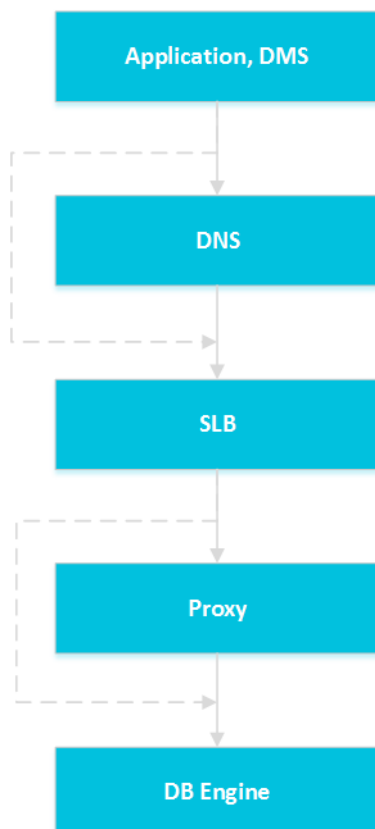
## 9.4 Features

### 9.4.1 Data link service

#### 9.4.1.1 Overview

The data link service allows you to add, delete, modify, and search data.

You can connect to the KVStore for Redis service by using your application.



### 9.4.1.2 DNS

The Domain Name System (DNS) module can dynamically resolve domain names to IP addresses. Therefore, IP address changes cannot affect the performance of KVStore for Redis.

For example, the domain name of a KVStore for Redis instance is `test.kvstore.aliyun.com`, and the IP address corresponding to this domain name is `10.1.1.1`.

You can connect to the KVStore for Redis instance if you add `test.kvstore.aliyun.com` or `10.1.1.1` to the connection pool of your application. If you migrate the KVStore for Redis instance to another host after a failure occurs or upgrades the instance version, the IP address may change to `10.1.1.2`. You can connect to the KVStore for Redis instance if you add `test.kvstore.aliyun.com` to the connection pool of your application. However, if you add `10.1.1.1` to the connection pool, you cannot connect to the instance.

### 9.4.1.3 SLB

The Server Load Balancer (SLB) module can forward traffic to available instance IP addresses. Therefore, physical server changes cannot affect the performance of KVStore for Redis.

For example, the private IP address of a KVStore for Redis instance is `10.1.1.1`. The IP address of the Proxy or DB Engine module is `192.168.0.1`. The SLB module forwards all traffic destined for `10.1.1.1` to `192.168.0.1`. When the Proxy or DB Engine module fails, the secondary Proxy or DB Engine module with the IP address `192.168.0.2` takes over for `192.168.0.1`. The SLB module redirects access traffic from `10.1.1.1` to `192.168.0.2` and the KVStore for Redis instance continues to run normally.

### 9.4.1.4 Proxy

The Proxy module provides some features such as data routing, traffic detection, and session persistence.

- **Data routing:** supports partition policies and complex queries for distributed routes based on a cluster architecture.
- **Traffic detection:** reduces the risks from cyberattacks that exploit Redis vulnerabilities.
- **Session persistence:** prevents connection interruptions in the case of failures.

### 9.4.1.5 DB Engine

KVStore for Redis supports standard Redis protocols of the corresponding engine versions as described in the following table.

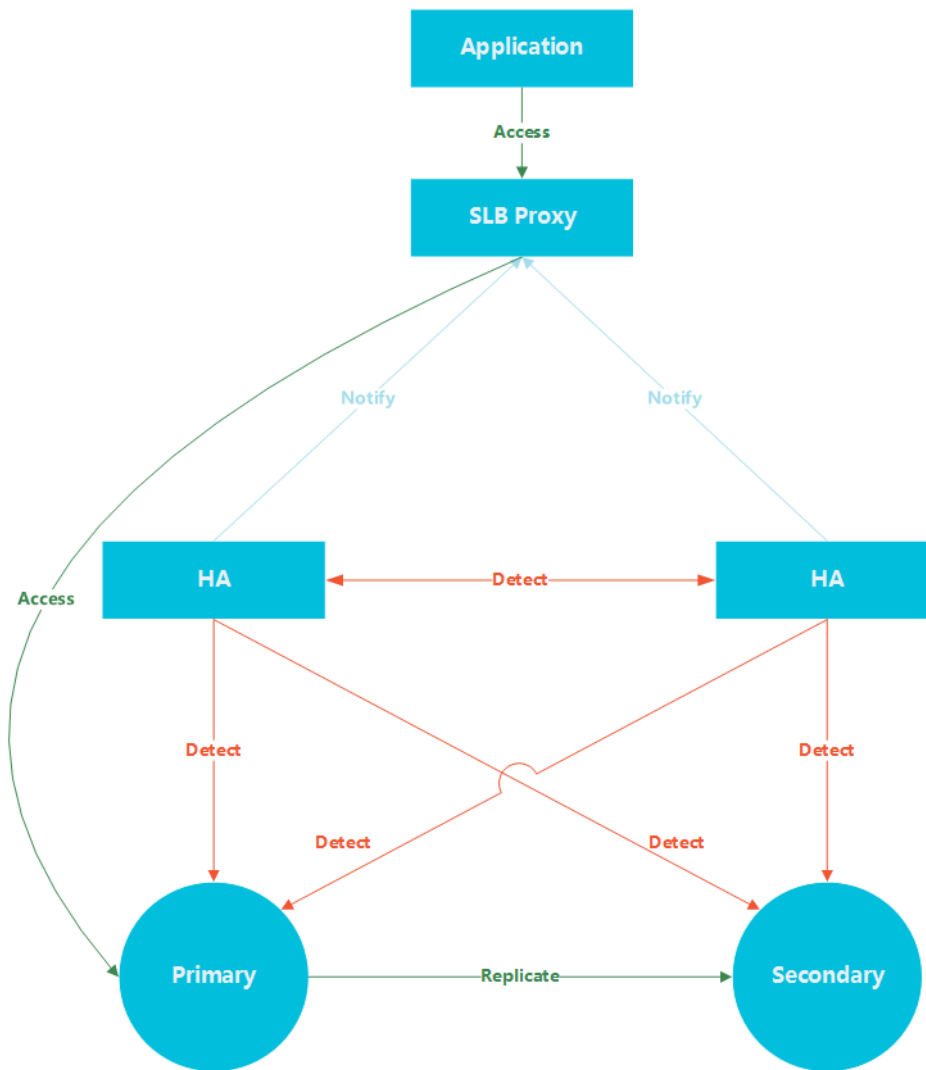
Engine	Version
Redis	Compatible with Redis 2.8 and Redis 3.0 GEO.
Redis	Redis 4.0

## 9.4.2 HA service

### 9.4.2.1 Overview

The high-availability (HA) service guarantees the availability of data link services and handles internal database exceptions.

The HA service is also highly available because this service contains multiple HA nodes.



### 9.4.2.2 Detection

The Detection module checks whether the primary and secondary nodes of the database engine are operating normally.

An HA node receives the heartbeat from the primary database engine node at an interval of 8 to 10 seconds. This information, combined with the heartbeat information of the secondary and other HA nodes, allows the Detection module to eliminate false negatives and positives caused by exceptions such as network jitter. As a result, switchover can be completed within 30 seconds.

### 9.4.2.3 Repair

The Repair module maintains replications between the primary node and the secondary node of DB Engine. This module also fixes errors that occur on either node during normal operations as follows:

- Automatically fixes exceptionally disconnected replications between these nodes
- 
- Automatically fixes table-level damages on both nodes.
- Automatically saves crash events and fixes the failures on both nodes.

#### 9.4.2.4 Notice

The Notice module notifies the SLB or Proxy module of status changes of primary and secondary nodes. Therefore, you can connect to available nodes.

For example, the Detection module locates an exception on a primary node and notifies the Repair module to fix the exception. If the Repair module fails to resolve the issue, the Repair module notifies the Notice module to perform failover. Afterward, the Notice module forwards the failover request the Server Load Balancer (SLB) or Proxy module to switch all traffic to the secondary node. Meanwhile, the Repair module creates a secondary node on a different physical server and synchronizes this change to the Detection module. The Detection module checks the health status of the instance again to verify that the instance is healthy.

### 9.4.3 Monitoring service

#### 9.4.3.1 Service-level monitoring

The independent Service module provides service-level monitoring. The Service module of KVStore for Redis monitors features, response time, and other metrics of other dependent cloud services such as Server Load Balancer (SLB), and checks whether these services run normally.

#### 9.4.3.2 Network-level monitoring

The Network module traces the network status. The monitoring metrics include:

- Connection conditions between ECS instances and KVStore for Redis instances.
- Connection conditions between physical servers of KVStore for Redis.
- Packet loss rates of routers and VSwitches.

#### 9.4.3.3 OS-level monitoring

The operating system (OS) module traces status of hardware and the kernel of an operating system. The monitoring metrics include:



- **Hardware inspection:** the OS module regularly checks the running status of devices such as CPUs, memory modules, motherboards, and storage devices. When locating any potential hardware failures, the module automatically raises a request for repair.
- **OS kernel monitoring:** the OS module traces all kernel requests for databases, and analyzes the cause of a slow or error response to a request according to the kernel status.

#### 9.4.3.4 Instance-level monitoring

The Instance module collects information of KVStore for Redis instances. The monitoring metrics include:

- Instance availability.
- Instance capacity.

#### 9.4.4 Scheduling service

The scheduling service allocates and integrates underlying resources of KVStore for Redis, so you can activate and migrate instances.

When you create an instance in the console, the scheduling service computes and selects the most suitable physical server to handle the traffic.

After long-term operations such as instance creation, deletion, and migration, a data center generates resource fragments. The scheduling service can calculate resource fragmentation in the data center and regularly initiates resource integration to improve service performance of the data center.

## 10 ApsaraDB for MongoDB

---

### 10.1 What is ApsaraDB for MongoDB?

ApsaraDB for MongoDB is a stable, reliable, and scalable database service fully compatible with MongoDB protocols. This service provides a full range of database solutions, such as disaster recovery, data backup, data recovery, monitoring, and alerts.

ApsaraDB for MongoDB uses the three-node replica set architecture by default. The primary node supports read/write access, the secondary node provides routine read-only operations, and the standby node is hidden to ensure high availability.

ApsaraDB for MongoDB supports multiple features to ensure the security and availability of services, including:

- Access control: database credential management and IP address whitelist
- Network isolation
- Data backup
- Version maintenance
- Service authorization

### 10.2 Benefits

High availability

- Three-node replica set high-availability architecture.

The ApsaraDB for MongoDB service uses a high-availability architecture that features a three-node replica set. The three data nodes are located on different physical servers and automatically synchronize data. The primary and secondary nodes provide services. When the primary node fails, the system automatically selects a new primary node. When the secondary node fails, the standby node takes over the services.

- Automatic backup with a single click.

Data is automatically backed up and uploaded to Object Storage Service (OSS) every day. This improves data disaster recovery capabilities while effectively

**reducing disk space consumption. The backup files can be used to restore the instance data to the original instance. This effectively prevents irreversible effects on service data caused by incorrect operations or other reasons.**

#### High security

- **Anti-DDoS:** This feature monitors traffic at the network ingress in real time. When heavy traffic is identified as an attack, traffic from the source IP addresses is scrubbed. If scrubbing is ineffective, the black hole mechanism is triggered.
- **IP whitelist configuration:** A maximum of 1,000 IP addresses are allowed to connect to an ApsaraDB for MongoDB instance, directly controlling risks at the source.

#### Ease of use

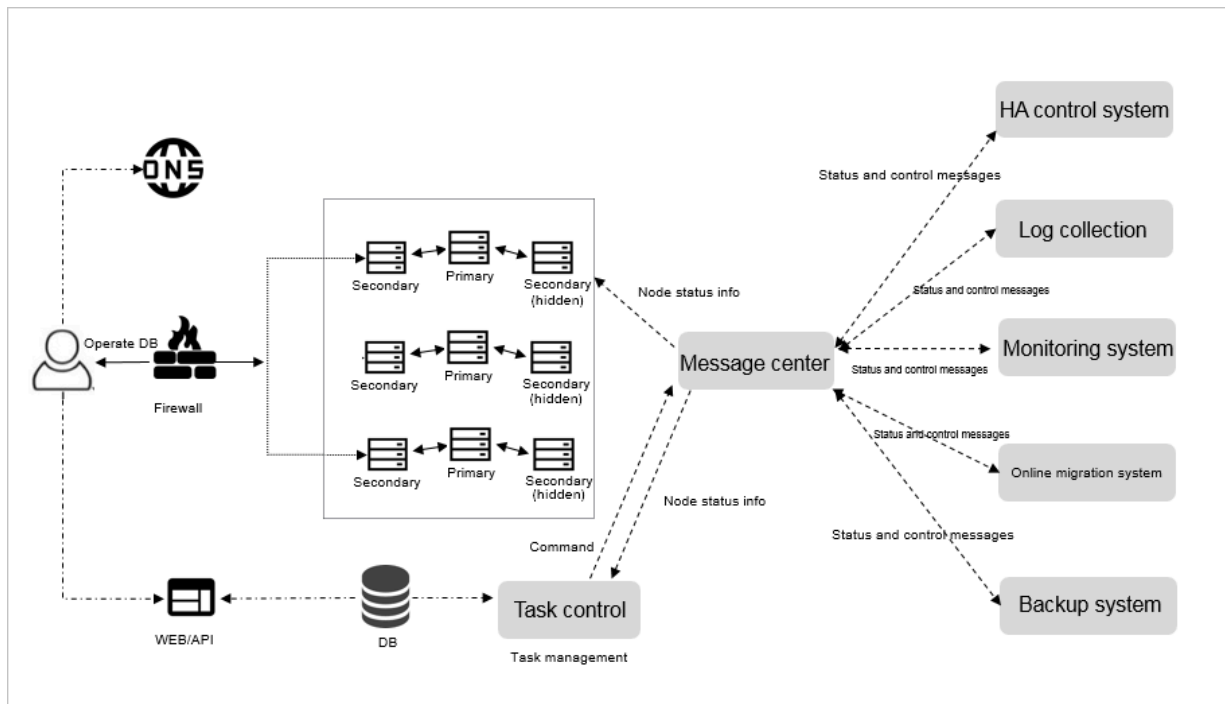
**ApsaraDB for MongoDB provides sound performance monitoring. It provides monitoring information about the CPU utilization, IOPS, connections, and disk capacity as well as real-time monitoring and alerting. It enables you to be aware of all instance statuses.**

#### Scalability

**ApsaraDB for MongoDB supports three-node replica sets that can be elastically scaled out. You can change the configuration of your instance if the current configuration is too high or is insufficient to meet the performance requirements of your application. The configuration change process is completely transparent and will not affect your business.**

## 10.3 Architecture

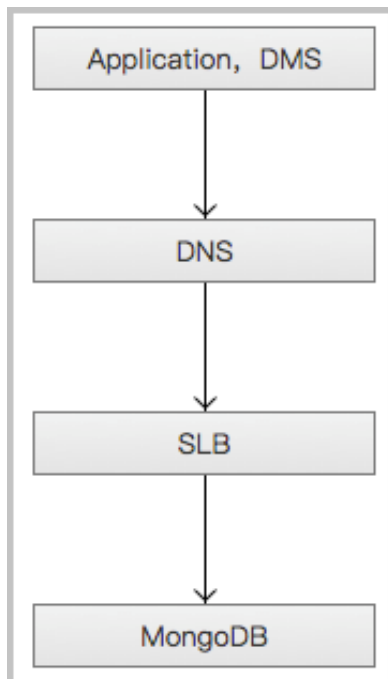
**ApsaraDB for MongoDB supports six core services: data link, scheduling, backup, high availability, monitoring, and migration.**



## 10.4 Features

### 10.4.1 Data link service

The data link service supports operations on data.



## DNS

For example, the domain name of an ApsaraDB for MongoDB instance is `mongodb.aliyun.com`, and the corresponding IP address is `10.1.1.1`. To connect an application to the instance, you can create a connection to URL `mongodb.aliyun.com` or IP address `10.1.1.1` in the connection pool.

However, the IP address may change to `10.1.1.2` if the instance is upgraded or migrated. In this case, if the connection in the connection pool has been created as `mongodb.aliyun.com`, the application can still access the instance. If the IP address is configured in the connection pool, the instance is no longer accessible.

## SLB

The SLB module provides instance IP addresses, including both internal and public IP addresses, to prevent physical server changes from affecting instance performance.

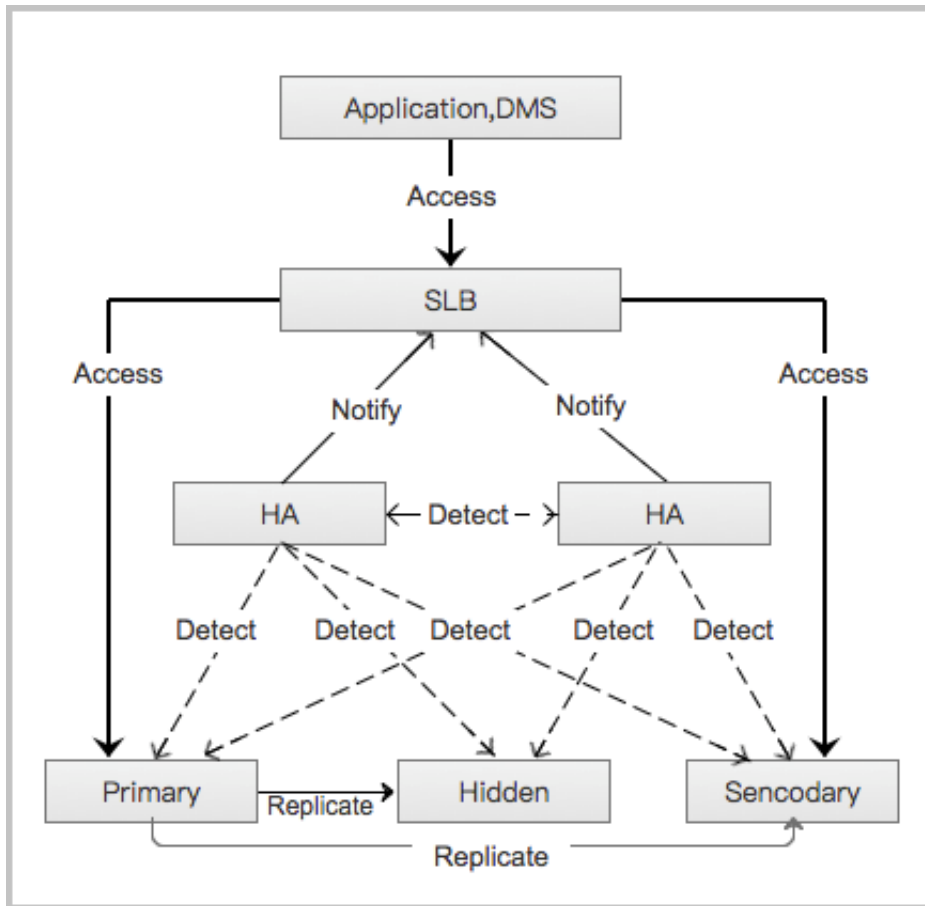
For example, the internal IP address of an ApsaraDB for MongoDB instance is `10.1.1.1`, and the corresponding ApsaraDB for MongoDB instance runs on a server whose IP address is `192.168.0.1`. Typically, the SLB module forwards all traffic destined for `10.1.1.1` to `192.168.0.1`.

If `192.168.0.1` fails, another server in the hot standby state with IP address `192.168.0.2` will take over for the server with IP address `192.168.0.1`. The SLB module then redirects all traffic destined for `10.1.1.1` to `192.168.0.2`.

### 10.4.2 High availability service

The high availability (HA) service guarantees the availability of data link services and handles internal database exceptions.

In addition, this service is based on multiple HA nodes that are also highly available .



### Detection

The Detection module detects the running or faulty status of the primary, secondary, and hidden nodes for ApsaraDB for MongoDB. An HA node uses heartbeat information, which is acquired at an interval of 8 to 10 seconds, to determine the health status of the primary node. This information, combined with the heartbeat information of the secondary and hidden nodes, allows the Detection module to eliminate any risk of false negatives and positives caused by exceptions such as network jitters. Switchover can be completed quickly.

### Repair

The Repair module maintains the replication relationship among the primary, secondary, and hidden nodes, and fixes faulty nodes or creates new nodes.

### Notice

The Notice module informs SLB of node status changes to ensure that you can access the available node.

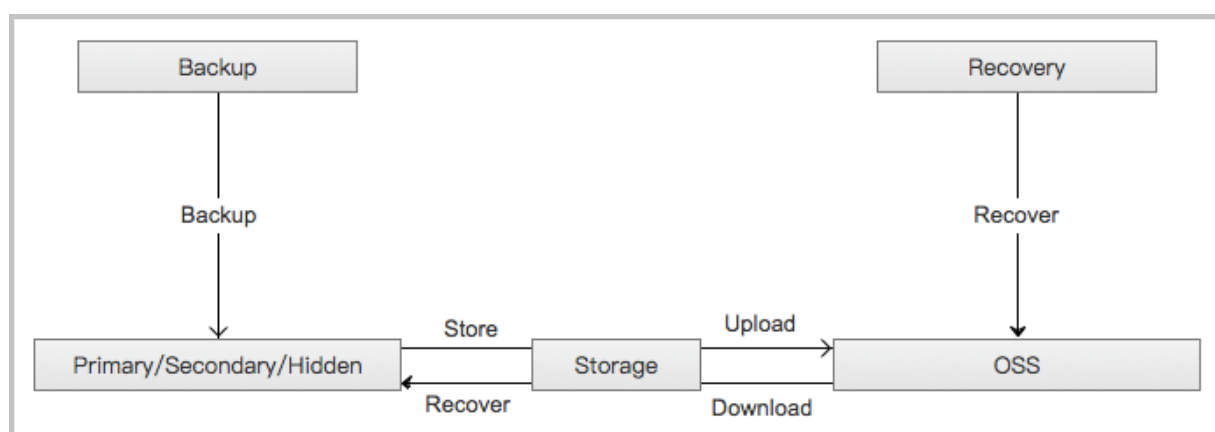
For example, the Detection module will instruct the Notice module to switch traffic if the Detection module discovers that an exception occurs with the primary

node. The Notice module then forwards the switched traffic request to SLB, which redirects traffic from the primary node to the secondary node or from the secondary node to the hidden node. In this circumstance, the secondary node becomes the primary node and the hidden node becomes the secondary node.

During this process, the Repair module attempts to fix the original primary node and convert it to a new hidden node. If the Repair module fails to fix the original primary node, the Repair module will create a new hidden node on another physical server and synchronize the change to the Detection module. The Detection module then incorporates the information and rechecks the health status of the instance.

### 10.4.3 Backup service

The backup service supports offline data backup, transfer, and recovery.



#### Backup

The Backup module backs up and compresses data and logs of an instance, and uploads the compressed files to OSS. Data backup in ApsaraDB for MongoDB is performed on the hidden node to avoid affecting services on the primary and secondary nodes.

#### Recovery

The Recovery module restores backup files stored in OSS to a specified node.

**Primary node rollback:** You can roll back the settings on the primary node to a specific point in time if you mistakenly perform operations on data.

**Secondary and hidden node restore:** The system automatically selects a new secondary node to reduce risks when an irreparable failure occurs with the original secondary node.

#### Storage

**The Storage module uploads, dumps, and downloads backup files. Currently, all backup data is uploaded to OSS for storage. You can obtain temporary links to download the data as needed.**

### 10.4.4 Monitoring service

**The monitoring service tracks the status of services, networks, operating systems, and instances.**

#### Service

**The Service module tracks the status of Alibaba Cloud services. For example, the Service module can monitor SLB, OSS, and SLS services and check whether their functions work as expected and the response time is acceptable. ApsaraDB for MongoDB is dependent on these services. The module also uses corresponding logs to check whether the internal services of ApsaraDB for MongoDB are running properly.**

#### Network

**The Network module tracks the status of networks. For example, the Network module can monitor the connectivity between ECS and ApsaraDB for MongoDB instances and among ApsaraDB for MongoDB physical machines. It can also monitor packet loss rates of VRouters and VSwitches.**

#### OS

**The OS module tracks the status of hardware and OS kernels.**

#### Examples:

- **Hardware inspection:** The OS module regularly checks the running status of components such as CPUs, memory modules, motherboards, and storage devices . If the module detects any potential hardware failures, it automatically submits a repair ticket.
- **OS kernel monitoring:** The OS module tracks all kernel invocations of databases and analyzes the cause of a slow or faulty invocation based on the kernel status.



## Instance

**The Instance module supports the following features:**

- **Collects ApsaraDB for MongoDB instance information.**
- **Provides instance availability information.**
- **Monitors instance capacity and performance metrics.**
- **Records statement executions for instances.**

## 10.4.5 Scheduling service

**The scheduling service allocates resources and manages instance versions.**

## Resource

**The Resource module allocates and integrates underlying ApsaraDB for MongoDB resources. This module allows you to create and modify instances.**

**For example, when you use the ApsaraDB for MongoDB console or API to create an instance, the Resource module will calculate and then select the most suitable server to handle the network traffic. After you have created, deleted, and migrated instances multiple times, the Resource module can calculate the resource fragmentation rate in a zone and periodically integrates the resources to improve service capabilities of the zone.**

## Version

**The Version module allows you to upgrade ApsaraDB for MongoDB instances. For example, you can upgrade an ApsaraDB for MongoDB instance to a major version, such as from version 3.2 to version 3.4. You can also upgrade an instance to a minor version that has optimized the source code or kernel as required.**

## 10.4.6 Migration service

**The migration service enables you to migrate data from a user-created database to ApsaraDB for MongoDB.**

**Data Transmission Service (DTS) is a data stream service provided by Alibaba Cloud for data exchanges between data sources. It supports full and incremental migration for ApsaraDB for MongoDB.**

- **Full data migration: DTS migrates all data from source databases to destination instances.**

- **Incremental data migration:** During incremental migration, the updated data in the local MongoDB database is synchronized to an ApsaraDB for MongoDB instance. Ultimately, the local MongoDB database and the ApsaraDB for MongoDB instance enter the dynamic synchronization process. Incremental migration enables data migration from a local MongoDB database to an ApsaraDB for MongoDB instance without interrupting the services provided by the local MongoDB database.

# 11 KVStore for Memcache

---

## 11.1 What is KVStore for Memcache?

**KVStore for Memcache is a memory-based cache service for high-speed access to large amounts of small-size data. KVStore for Memcache can reduce the load on back-end storage services and speed up website and application responses.**

**KVStore for Memcache supports data in the key-value structure. It can communicate with memcached-compatible clients.**

**KVStore for Memcache supports out-of-the-box deployment. It also relieves the load on databases from dynamic Web applications and improves website response speed by using the cache service.**

**Similar to user-created memcached databases, KVStore for Memcache is also compatible with the memcached protocol and user environments. The difference is that the data, hardware infrastructure, network security, and system maintenance services used by KVStore for Memcache are all deployed on the cloud.**

### 11.1.1 Scenarios

Frequently-accessed businesses

**Frequently-accessed businesses include social networks, e-businesses, games, and advertisements. Frequently-accessed data can be stored in KVStore for Memcache, while underlying data can be stored in RDS.**

Large promotion businesses

**Large promotional events and flash sales place systems under high access pressure. Average databases cannot withstand such high read/write pressure, so KVStore for Memcache can act as a viable alternative.**

Inventory systems with counters

**ApsaraDB for RDS and KVStore for Memcache can be used in combination. RDS stores the specific data and database fields store the specific statistics. KVStore for Memcache reads the statistics, while RDS stores the statistics.**

## Data analysis businesses

**KVStore for Memcache can be used in combination with MaxCompute to analyze and process big data in a distributed manner. It is suitable for big data processing scenarios such as business analysis and data mining. The Data Integration service can simplify data operations by synchronizing data between KVStore for Memcache and MaxCompute.**

## 11.2 Benefits

### Ease of use

- **Out-of-the-box deployment:** Instances are available immediately after creation, facilitating fast business deployment.
- **Compatible with open-source memcached:** KVStore for Memcache is compatible with the memcached binary protocol. All clients that support this protocol and SASL can connect to KVStore for Memcache.
- **Visualized management and monitoring panel:** The console provides several monitoring metrics to facilitate management for Memcache instances.

### Cluster features

**KVStore for Memcache supports super large capacity and provides super high performance. The cluster output utilizes super large cluster instances to meet demands for large capacity and high performance.**

### Elastic scalability

- **Scale-out of storage capacity with a single click:** You can adjust the storage capacity of an instance in the console based on your business requirements.
- **Online scale-out without service interruption:** You can adjust the instance capacity without suspending your services or affecting your business.

### Resource isolation

**Instance-level resource isolation provides enhanced stability for individual services.**

### High security and reliability

- **Password authentication ensures secure and reliable access.**

- **Persistent data storage:** The use of memory and hard disks can provide high-speed data reading and writing and meet data persistence demands.

High availability

- **Each instance has a primary node and a secondary node.** This prevents service interruption caused by single point of failures (SPOFs).
- **Automatic detection and recovery of hardware faults:** KVStore for Memcache automatically detects hardware faults and fails services over within seconds to recover services.

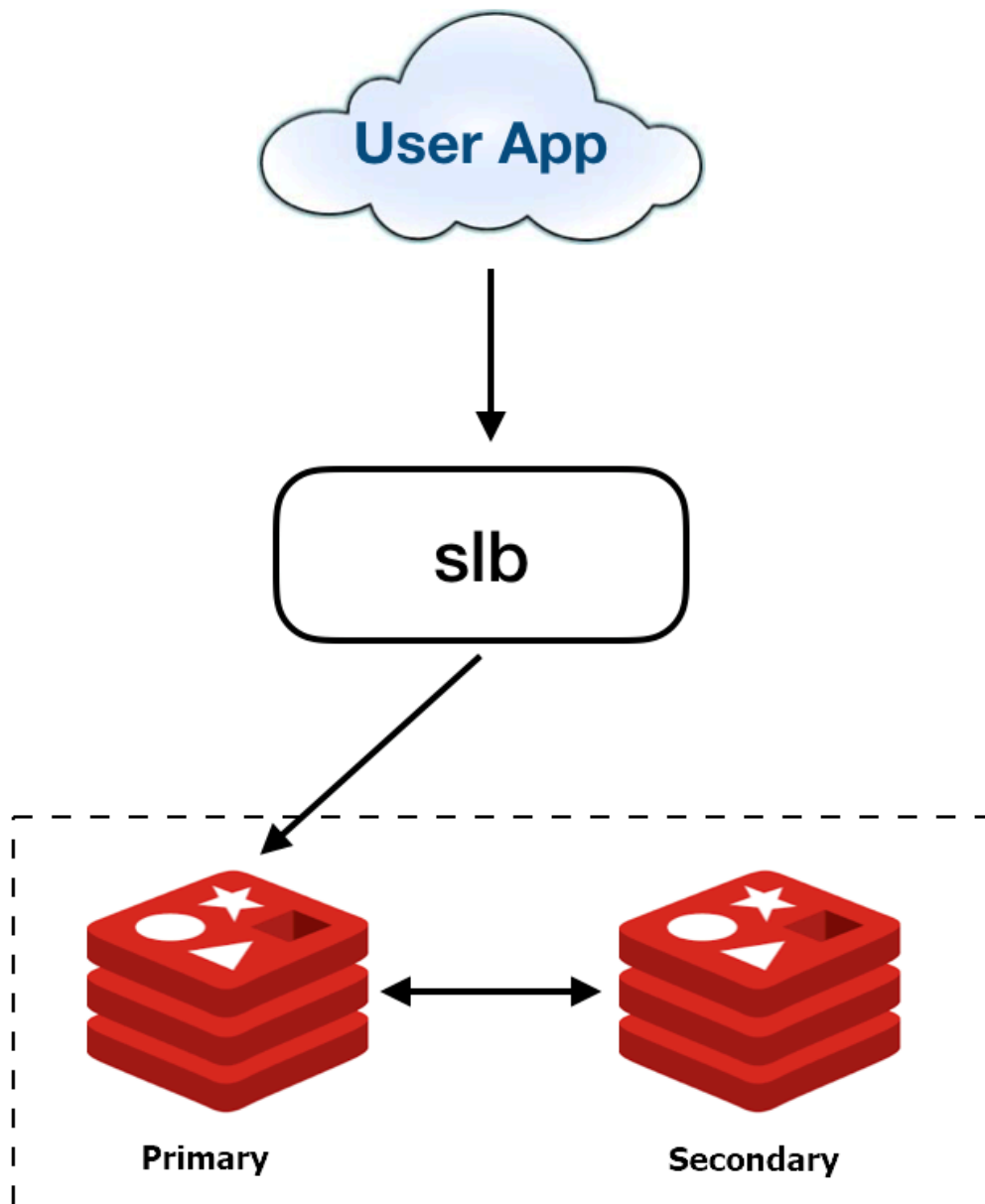
## 11.3 Architecture

### 11.3.1 Overall system architecture

**KVStore for Memcache supports the Memcache protocol based on the open-source Redis kernel. The system architecture is centered around the Redis kernel. There are two system architecture modes: primary/secondary and cluster.**

Primary/secondary architecture

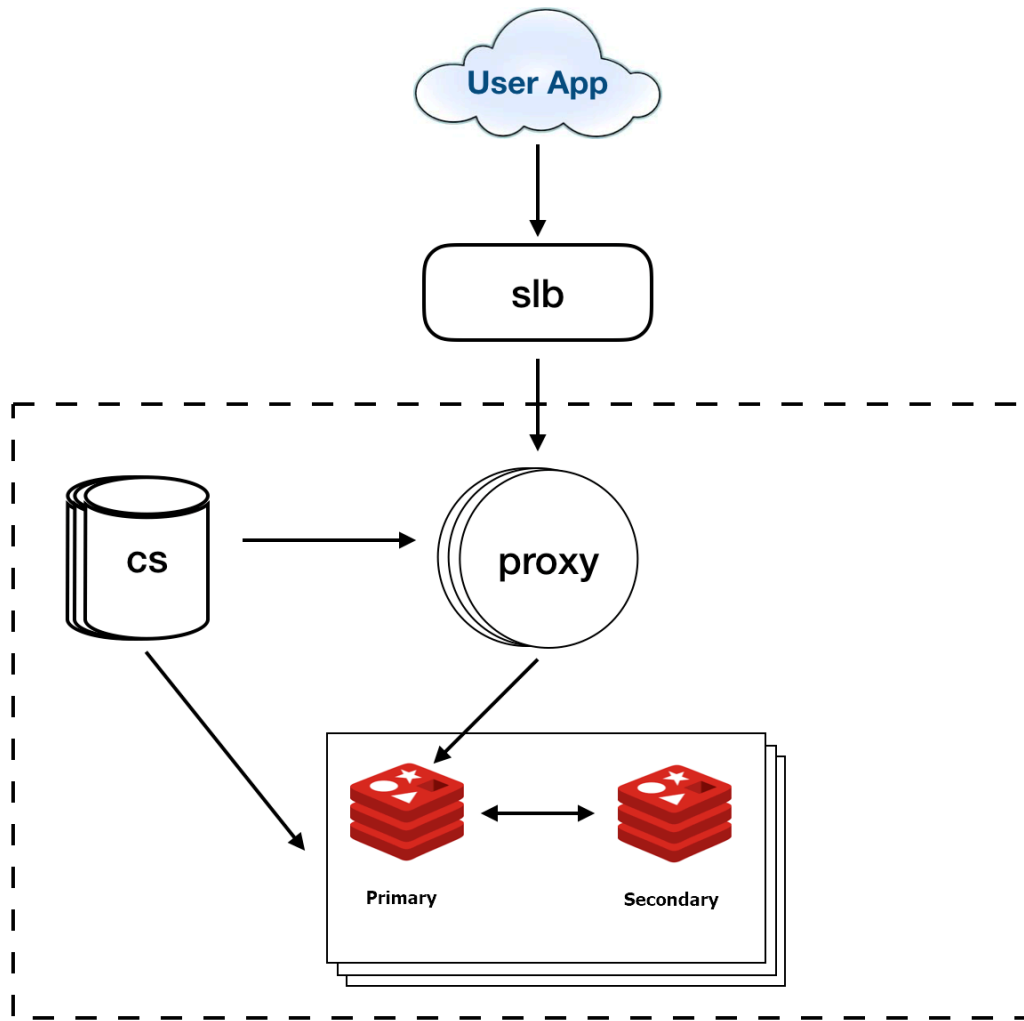
**The following figure shows the primary/secondary architecture.**



**You can directly access Redis through an SLB connection. The Redis kernel supports the Memcache protocol.**

Cluster architecture

**The following figure shows the cluster architecture.**



The cluster architecture is composed of three components: redis-config (cs), redis-proxy (proxy), and Redis.

One cluster architecture consists of multiple cs nodes, proxy nodes, and primary /secondary Redis nodes. After you access the proxy component through an SLB connection, the proxy component forwards request routes to a shard of the primary Redis database. The proxy component supports the memcached protocol.

### 11.3.2 System components and technical principles

This topic describes the components and technical principles of KVStore for Memcache.

#### Redis-config component

The redis-config (cs) component stores the metadata and topology information of the cluster, and performs cluster operations and maintenance. The cs component

**synchronizes the metadata and topology information of the cluster with the redis and proxy components.**

Redis-proxy component

**The redis-proxy (proxy) component is used by the client to connect to the Redis server and can implement the memcache protocol. The proxy component can authenticate user identities, forward request routes, provide slow and audit logs, and collect monitoring data every several minutes.**

Redis component

**The redis component is performance optimized and contains features developed by Alibaba Cloud based on the open-source Redis kernel to implement the memcache protocol.**

## 11.4 Features and principles

### 11.4.1 Data link service

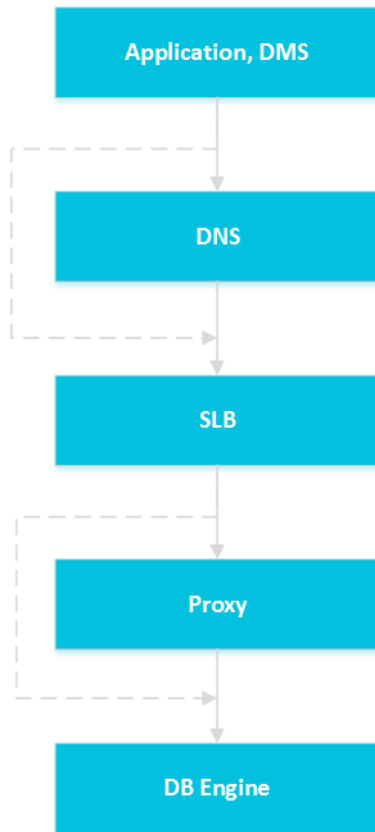
#### 11.4.1.1 Overview

**The data link service allows you to add, delete, modify, and query data.**



**You can connect to the KVStore for Memcache service through applications or through the graphical Data Management Service (DMS).**

Figure 11-1: Data link service



#### 11.4.1.2 DNS

**The DNS module dynamically resolves domain names into IP addresses to prevent IP address changes from affecting the performance of instances.**

**For example, assume that the domain name of a KVStore for Memcache instance is `test.kvstore.aliyun.com`, and the IP address corresponding to this domain name is `10.1.1.1`. Either `test.kvstore.aliyun.com` or `10.1.1.1` can be configured in the connection pool of a program and used to connect to the KVStore for Memcache instance. If the KVStore for Memcache instance is uploaded or is migrated to another host due to failover, the IP address may change to `10.1.1.2`. In this case, the KVStore for Memcache instance can be accessed through `test.kvstore.aliyun.com` as configured in the connection pool, but cannot be accessed through the configured address `10.1.1.1`.**

### 11.4.1.3 SLB

The SLB module provides instance IP addresses to prevent host changes from affecting the performance of instances.

For example, assume that the internal IP address of a KVStore for Memcache instance is 10.1.1.1, and the IP address of the proxy or DB engine is 192.168.0.1. Typically, the SLB module would forward all traffic destined for 10.1.1.1 to 192.168.0.1.

If the proxy or DB engine fails, the secondary proxy or DB engine with IP address 192.168.0.2 is activated. The SLB module then forwards all traffic destined for 10.1.1.1 to 192.168.0.2, which ensures that the instance can continue to provide services uninterrupted.

### 11.4.1.4 Proxy

The Proxy module provides a number of functions including data routing, traffic detection, and session persistence.

- **Data routing:** supports partition policies and complex queries for distributed routes based on the cluster architecture.
- **Traffic detection:** reduces the risks from network attacks that exploit vulnerabilities in KVStore for Memcache.
- **Session persistence:** resolves database disconnection issues.

### 11.4.1.5 DB engine

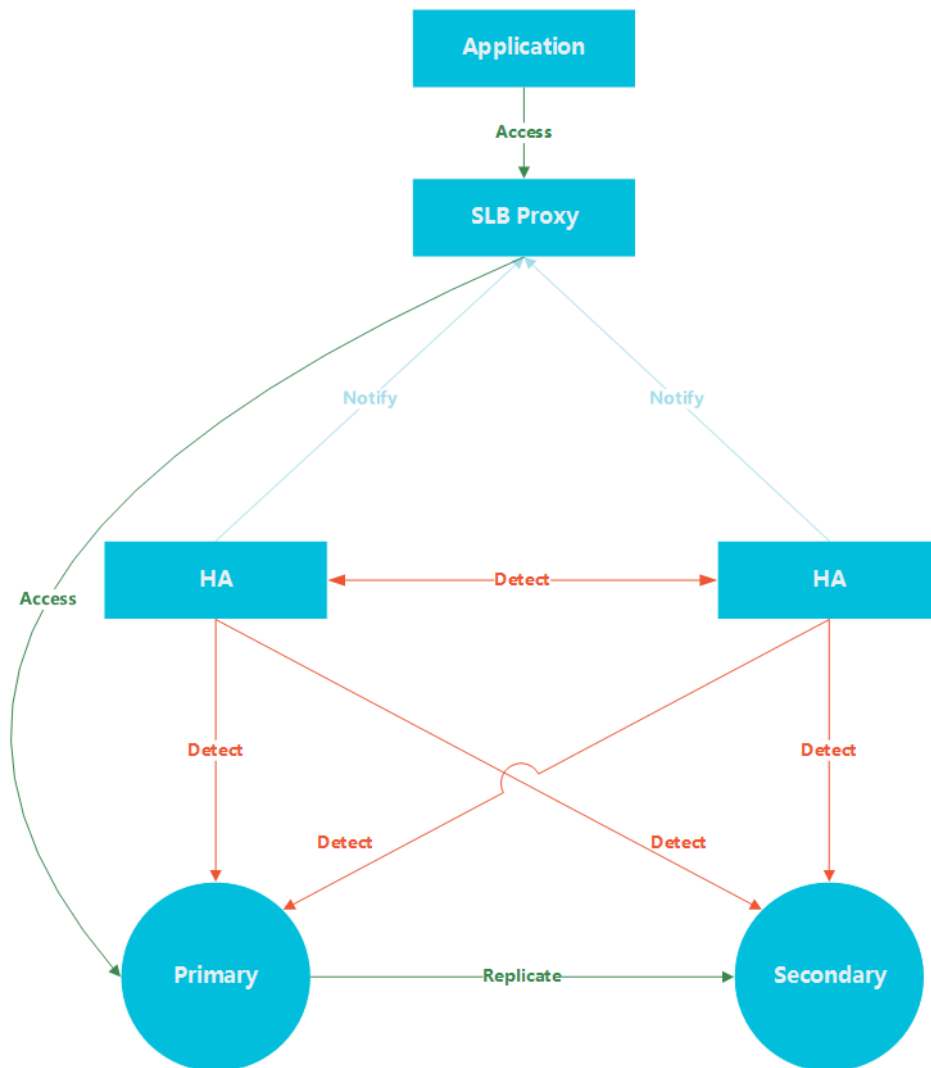
The DB engine of KVStore for Memcache supports the standard memcached text protocol and binary protocol. The DB engine can be directly connected to a number of different clients.

## 11.4.2 High availability service

### 11.4.2.1 Overview

The high availability (HA) service guarantees the availability of data link services and handles internal database exceptions. The HA service itself is also highly available because multiple HA nodes are used to implement the service.

Figure 11-2: High availability service



### 11.4.2.2 Detection

The Detection module checks whether the primary and secondary nodes of the DB engine are operating normally.

An HA node receives the heartbeat from the primary database engine node at an interval of 8 to 10 seconds. This information, combined with the heartbeat information of the secondary and other HA nodes, allows the Detection module to

eliminate false negatives and positives caused by exceptions such as network jitter. As a result, switchover can be completed within 30 seconds.

### 11.4.2.3 Repair

The Repair module maintains the replication between the primary and secondary nodes of the DB engine. It can also correct any errors that occur on either node during normal operations.

- Automatically fixes exceptionally disconnected replications between two nodes.
- Automatically fixes damaged tables on either node.
- Automatically saves the crash events of the primary or secondary node and fixes the failures.

### 11.4.2.4 Notice

The Notice module informs the SLB or Proxy module of status changes in the primary and secondary nodes to ensure continued access to the correct node.

For example, the Detection module can discover problems with the primary node and instructs the Repair module to fix these problems. If the Repair module cannot resolve a problem, it instructs the Notice module to switch traffic to the secondary node. The Notice module forwards the switching request to the SLB or Proxy module, which will then redirect all traffic to the secondary node. At the same time, the Repair module creates a new secondary node on a different host and synchronizes this change back to the Detection module. The Detection module rechecks the health status of the instance.

## 11.4.3 Monitoring service

### 11.4.3.1 Service-level monitoring

The Service module tracks the status of services. For example, the Service module can monitor the operating status of related cloud services such as SLB.

### 11.4.3.2 Network-level monitoring

The Network module tracks the status of networks.

For example, the Network module can monitor the connectivity between ECS and KVStore for Memcache instances, or between KVStore for Memcache physical servers. The module can also monitor packet loss rates on routers and VSwitches.

### 11.4.3.3 OS-level monitoring

The OS module tracks the statuses of hardware and the OS kernel. The information monitored by the OS module includes:

- **Hardware maintenance:** The OS module monitors the operating status of the CPU , memory, motherboard, and storage device, predicts faults, and automatically submits repair reports when it predicts that a fault is imminent.
- **OS kernel status:** The OS module tracks all database calls, and analyzes the causes of slow calls or call errors based on the kernel status.

### 11.4.3.4 Instance-level monitoring

The Instance module collects instance-level information for KVStore for Memcache , such as instance availability and capacity.

## 11.4.4 Scheduling service

The scheduling service integrates and allocates the underlying resources of KVStore for Memcache to facilitate instance creation and migration.

For example, if you use the console to create an instance, the scheduling service will select the optimal physical server to handle the instance traffic.

After instance creation, deletion, and migration operations are performed many times over a long period, resource fragments are generated in the data center. The scheduling service calculates the degree of resource fragmentation and periodically integrates resources to increase the service capacity of the data center.

## 12 AnalyticDB for PostgreSQL

---

### 12.1 What is AnalyticDB for PostgreSQL?

AnalyticDB for PostgreSQL (formerly known as HybridDB for PostgreSQL) is a distributed cloud database that is composed of multiple compute groups to provide Massively Parallel Processing (MPP) data warehousing service.

AnalyticDB for PostgreSQL is developed based on the Greenplum Open Source Database project and has been enhanced by Alibaba Cloud. This service includes the following features:

- **Compatible with Greenplum.** You can use any tool that is compatible with Greenplum.
- **Supports OSS, JSON data format, and HyperLogLog, a probability cardinality estimation algorithm.**
- **Complies with SQL 2008 standard query syntax and OLAP aggregate functions, providing flexible hybrid analysis capabilities.**
- **Supports hybrid storage with row store and column store, enhancing analytics performance.**
- **Supports data compression to reduce storage costs.**
- **Provides online scaling and performance monitoring services to help reduce O&M burden. This enables DBAs, developers, and data analysts to focus on improving enterprise productivity and creating core business values with SQL.**

#### 12.1.1 Scenarios

AnalyticDB for PostgreSQL is applicable to the following OLAP data analysis services.

- **ETL for offline data processing**

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

- Supports standard SQL, OLAP window functions, and stored procedures.
- Provides the CASCADE-based SQL optimizer to make complex queries without the need for tuning.
- Uses the MPP architecture that can be horizontally scaled and can process petabytes of data in seconds.
- Provides column store-based high-performance storage and aggregation of large tables and high compression ratio to save storage space.

- **Online high-performance query**

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

- Allows you to write and update high-throughput data through INSERT, UPDATE, and DELETE operations.
- Allows you to query data based on row store and multiple indexes (B-tree, bitmap, and hash) to obtain results in milliseconds.
- Supports distributed transactions, standard database isolation levels, and HTAP.

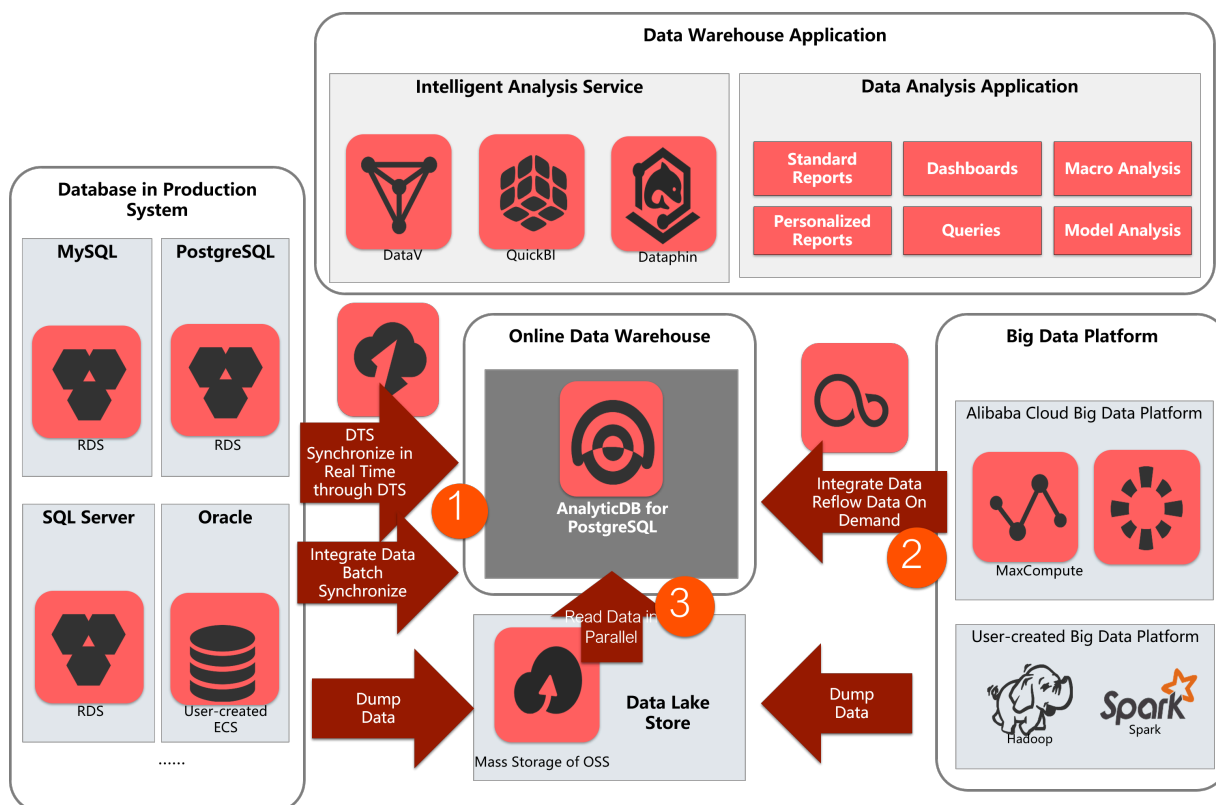
- **Multi-model data analysis**

AnalyticDB for PostgreSQL provides the following benefits for processing a variety of unstructured data sources:

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

Typical scenarios

AnalyticDB for PostgreSQL is applicable to the three following scenarios:



- **Data warehousing service**

Data Transmission Service (DTS) can synchronize data in real time in production system databases such as ApsaraDB RDS for MySQL, ApsaraDB RDS for PostgreSQL, ApsaraDB for POLARDB, and traditional databases such as Oracle and SQL Server. Data can also be synchronized in batches to AnalyticDB for PostgreSQL through the data integration service (DataX). AnalyticDB for PostgreSQL supports Extract, Transform, and Load (ETL) operations on large amounts of data. You can also use DataWorks to schedule these tasks. AnalyticDB for PostgreSQL also provides high-performance online analysis capabilities and can use Quick BI, DataV, Tableau, and FineReport for report presentation and real-time query.

- **Big data analytics platform**





To perform high-performance analysis, processing, and exploration, you can import huge amounts of data from MaxCompute, Hadoop, and Spark to AnalyticDB for PostgreSQL through DataX or OSS.




- Data lake analytics

AnalyticDB for PostgreSQL can use an external table mechanism to access the huge amounts of data stored in OSS in parallel and build an Alibaba Cloud data lake analytics platform.

## 12.2 Benefits

 <b>Real-time analysis</b>	<p>Supports SQL syntax for distributed real-time analysis of GIS-based geographic data, integrating with Internet of Things (IoT) and the Internet to provide location-based services (LBS).</p> <p>Supports SQL syntax for distributed real-time analysis of JSON data, XML data, and fuzzy strings. This is ideal for financial, governmental, and enterprise customers who want to process packets and sentiment analysis.</p>
 <b>Stability and reliability</b>	<p>Provides ACID properties for distributed transactions. Transactions are consistent in different nodes and all data is synchronized in primary and secondary nodes.</p> <p>Supports distributed deployment and provides compute group, server, and rack protection to secure your data infrastructure.</p>
 <b>Easy to use</b>	<p>Supports a large number of OLAP SQL syntax, functions, and Oracle functions, and allows you to use popular BI software online.</p> <p>Supports communication between ApsaraDB RDS for PostgreSQL and ApsaraDB RDS for PPAS to provide an OLTP+OLAP hybrid transactional/analytical processing (HTAP).</p>
 <b>Ultra-high performance</b>	<p>Supports hybrid storage mode between row store and column store. During OLAP analysis, the performance of column store is much higher than that of row store.</p>

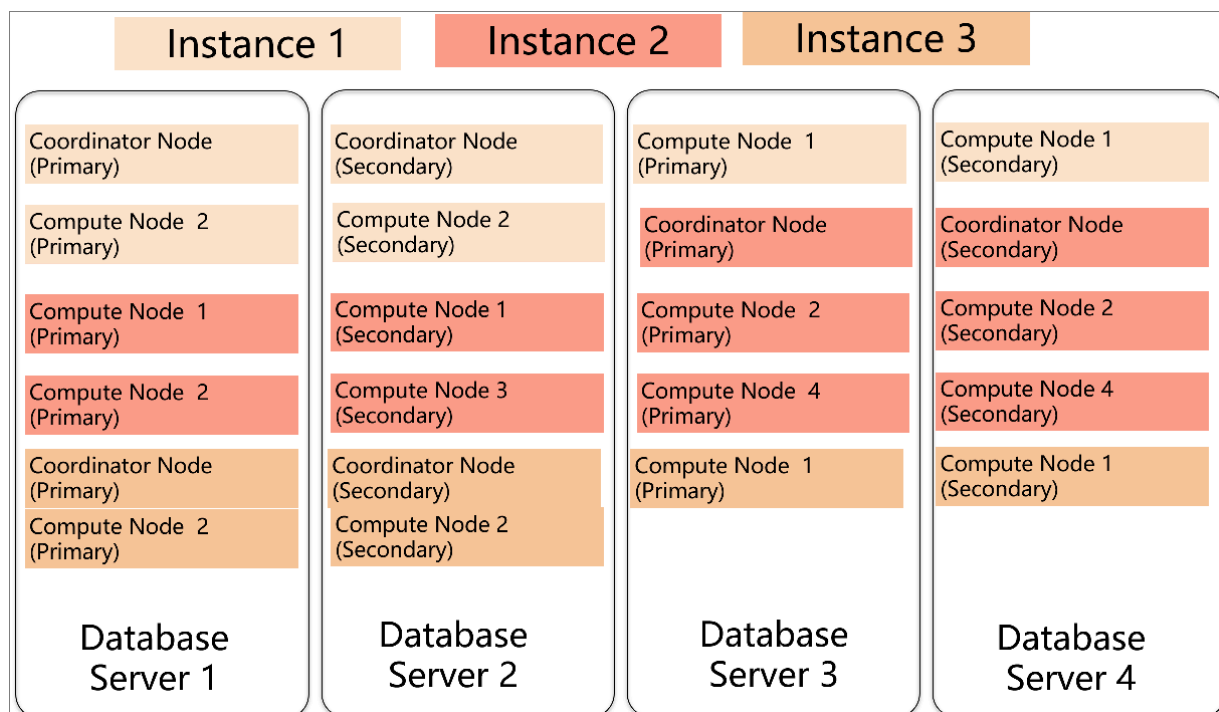
	<b>Supports high-performance parallel import of data from OSS, which has no limits compared with importing data through a single channel.</b>
 <b>Scalability</b>	<p><b>Enables you to scale up compute groups, CPU, memory, and storage resources on demand to improve OLAP performance.</b></p> <p><b>Supports transparent OSS operations. OSS offers a larger storage capacity for cold data that does not require online analysis.</b></p>

## 12.3 Architecture

Physical cluster architecture

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

Figure 12-1: Physical cluster architecture



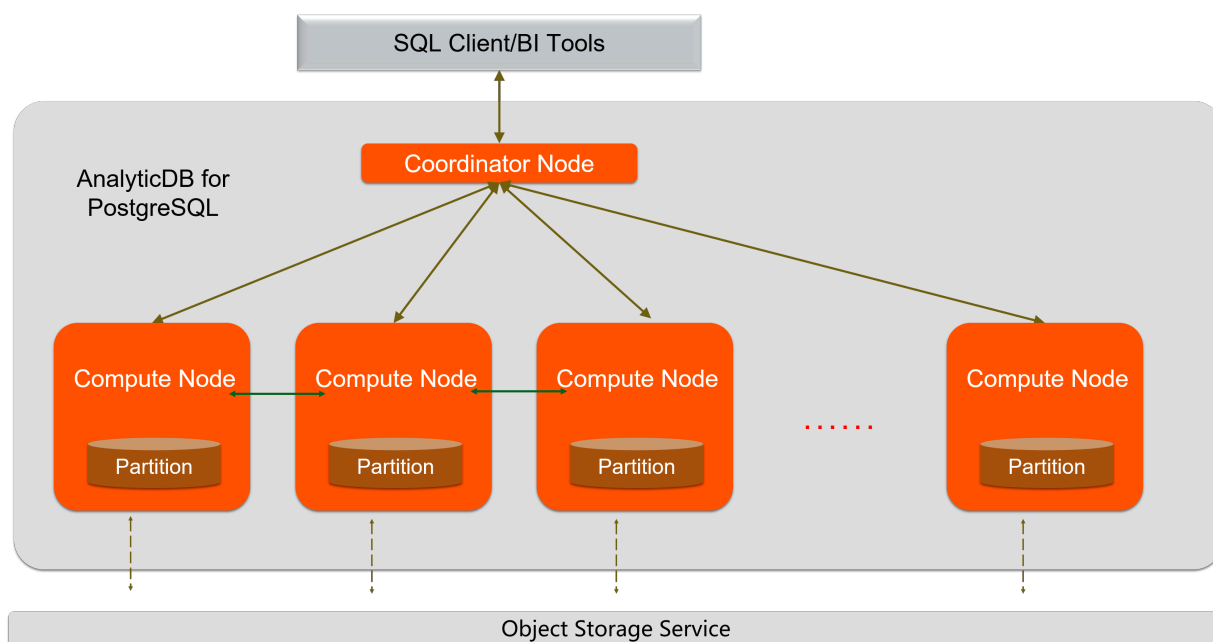
**You can create multiple instances in a physical cluster of AnalyticDB for PostgreSQL. Each cluster includes two components: the master and the segment.**

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

Logical architecture of an instance

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

Figure 12-2: Logical architecture of an instance



Data is distributed among segments by hash value or RANDOM function of a specified distributed column. Each segment consists of a primary node and a secondary node to ensure dual-copy storage. High-performance network communication is supported among nodes. When the master receives a request from the application, the master parses and optimizes SQL statements to generate a distributed execution plan. After the master sends the execution plan to the segments, the segments perform an MPP execution of the plan.

## 12.4 Features

### Distribution

- **MPP architecture**

AnalyticDB for PostgreSQL is based on the Massively Parallel Processing (MPP) architecture. The storage is extended linearly and computing capabilities are enhanced by adding more compute groups, which leverage the OLAP computing performance of each compute group.

- **Distributed transactions**

Supports distributed SQL OLAP and window functions, distributed PL/pgSQL stored procedures and triggers, and enables databases to support distributed computing.

### Learning and analysis

- **MADlib machine learning**

Provides a large number of SQL-based machine learning tools for data science users and is built in with more than 50 machine learning algorithms.

- **GIS-based geographic analysis**

Supports hybrid geographic data analysis that complies with the OpenGIS specifications, and enables you to use a single SQL statement to analyze a large amount of geographic data, such as population flow, area statistics, and traces.

### Data interconnection

- **Heterogeneous data import**

Imports data from MySQL databases by using the mysql2pgsql tool. You can use popular ETL tools to import data to AnalyticDB databases through the ETL process.

- **OSS heterogeneous data storage**

Uses standard SQL syntax to query format files stored in OSS by using external tables in real time.

- **Transparent data replication**

Replicates data transparently from ApsaraDB RDS for PostgreSQL or ApsaraDB RDS for PPAS without the need to program for consecutive incremental

**replication. This feature simplifies maintenance, and allows high-performance internal modeling and data cleansing for the imported data.**

#### Security

- **IP address whitelist**

**Allows you to add up to 1,000 IP addresses to the whitelist of AnalyticDB for PostgreSQL. This feature allows you to control risks from sources of access.**

- **Anti-DDoS**

**Monitors inbound traffic in real time, scrubs large amounts of malicious traffic by filtering source IP addresses, and throws affected servers into a black hole.**

#### Limits

- **For limits on core features of Greenplum Database, see [Summary of Greenplum Features](#).**
- **Permission limits:** The initial users, also known as root users, of AnalyticDB for PostgreSQL have the CREATEDB and CREATEROLE permissions. Root users do not have superuser permissions to perform certain operations such as executing `pg_ls_dir` and other file operation functions. Superusers have permissions to view and modify data of other non-superusers and kill their connections.
- **Does not support the PL/R and PL/Java extensions.**
- **Supports creating extensions with PL/Python but does not support creating functions with PL/Python.**
- **Does not support the gpfdist program.**
- **Does not support MapReduce interfaces, gphdfs interfaces, or on-premises external tables.**
- **Does not support automatic backup and restoration. AnalyticDB for PostgreSQL saves two copies of data. You can also use the `pg_dump` utility to back up data.**

### 12.4.1 Distributed architecture

**AnalyticDB for PostgreSQL is based on the MPP architecture. Data is distributed evenly among nodes by hash value or RANDOM function, and is analyzed and computed among nodes in parallel. The storage and computing capacities are scaled horizontally as more nodes are added. This ensures a quick response when the data volume increases.**

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

## 12.4.2 High-performance data analysis

AnalyticDB for PostgreSQL supports column store and row store for tables. Row store provides high update performance and column store provides high OLAP aggregate analysis performance for tables. AnalyticDB for PostgreSQL supports the B-tree index, bitmap index, and hash index that enable high-performance analysis, filtering, and query.

AnalyticDB for PostgreSQL adopts the CASCADE-based SQL optimizer. AnalyticDB for PostgreSQL combines the cost-based optimizer (CBO) and 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.

## 12.4.3 High-availability service

AnalyticDB for PostgreSQL builds a system for automatic monitoring, diagnosis, and error handling based on the Apsara platform of Alibaba Cloud, which helps to reduced O&M costs.

The master stores database metadata and receives query requests from clients to compile and optimize SQL statements. The master adopts a primary/secondary architecture to ensure strong consistency of metadata. If the primary master fails, the service is automatically switched to the secondary master.

All segments adopt a primary/secondary architecture to ensure strong data consistency between primary and secondary nodes when data is written into or updated. If the primary segment fails, the service is automatically switched to the secondary segment.

## 12.4.4 Data synchronization and tools

You can use Data Transmission Service (DTS) or DataWorks to synchronize data from MySQL or PostgreSQL databases to AnalyticDB for PostgreSQL. Popular extract, transform, and load (ETL) tools can import ETL data and schedule jobs on AnalyticDB for PostgreSQL databases. You can also use standard SQL syntax to query data from formatted files stored in OSS by using external tables in real time.

AnalyticDB for PostgreSQL supports Business Intelligence (BI) reporting tools, including Quick BI, DataV, Tableau, and FineReport. It also supports ETL tools, including Informatica and Kettle.

### 12.4.5 Data security

AnalyticDB for PostgreSQL supports IP whitelist configuration. You can add IP addresses of up to 1,000 servers that are allowed to access your instance to the whitelist. This enables you to control risks from the access source. AnalyticDB for PostgreSQL also supports Anti-DDoS that monitors inbound traffic in real time. When a large amount of malicious traffic is identified, it scrubs traffic through IP filtering. If traffic scrubbing is ineffective, it triggers the black hole process.

### 12.4.6 Supported SQL features

- Supports row store and column store.
- Supports multiple indexes, including the B-tree index, bitmap index, and hash index.
- Supports distributed transactions and standard isolation levels, which ensure data consistency among nodes.
- Supports character, date, and arithmetic functions.
- Supports stored procedures, user-defined functions (UDF), and triggers.
- Supports views.
- Supports range partitioning, list partitioning, and the definition of multi-level partitions.
- Supports multiple data types. The following table provides a list of data types and their information.

Data type	Alias	Storage	Range	Description
<b>bigint</b>	<b>int8</b>	8 bytes	-9223372036854775808 to 9223372036854775807	Large-range integer
<b>bigserial</b>	<b>serial8</b>	8 bytes	1 to 9223372036854775807	Large auto-increment integer
<b>bit [ (n) ]</b>	N/A	n bits	Bit string constant	Fixed-length bit string

Data type	Alias	Storage	Range	Description
<b>bit varying [ (n) ]</b>	<b>varbit</b>	<b>Variable-length bit string</b>	<b>Bit string constant</b>	<b>Variable-length bit string</b>
<b>boolean</b>	<b>bool</b>	<b>1 byte</b>	<b>true/false, t/f, yes/no, y/n, 1/0</b>	<b>Boolean value (true/false)</b>
<b>box</b>	<b>N/A</b>	<b>32 bytes</b>	<b>((x1,y1),(x2,y2))</b>	<b>A rectangular box on a plane, not allowed in distribution key columns</b>
<b>bytea</b>	<b>N/A</b>	<b>1 byte + binary string</b>	<b>Sequence of octets</b>	<b>Variable-length binary string</b>
<b>character [ (n) ]</b>	<b>char [ (n) ]</b>	<b>1 byte + n</b>	<b>String up to n characters in length</b>	<b>Fixed-length, blank-padded string</b>
<b>character varying [ (n) ]</b>	<b>varchar [ (n) ]</b>	<b>1 byte + string size</b>	<b>String up to n characters in length</b>	<b>Variable length with limit</b>
<b>cidr</b>	<b>N/A</b>	<b>12 or 24 bytes</b>	<b>N/A</b>	<b>IPv4 and IPv6 networks</b>
<b>circle</b>	<b>N/A</b>	<b>24 bytes</b>	<b>&lt;(x,y),r&gt; (center and radius)</b>	<b>A circle on a plane, not allowed in distribution key columns</b>
<b>date</b>	<b>N/A</b>	<b>4 bytes</b>	<b>4713 BC to 294,277 AD</b>	<b>Calendar date (year, month, day)</b>
<b>decimal [ (p, s) ]</b>	<b>numeric [ (p, s) ]</b>	<b>variable</b>	<b>No limit</b>	<b>User-specified precision, exact</b>
<b>double precision</b>	<b>float8</b>	<b>8 bytes</b>	<b>15 decimal digits of precision</b>	<b>Variable precision, inexact</b>
	<b>float</b>			



Data type	Alias	Storage	Range	Description
<b>inet</b>	N/A	12 or 24 bytes	N/A	IPv4 and IPv6 hosts and networks
<b>integer</b>	int or int4	4 bytes	-2.1E+09 to +2147483647	Typical choice for integer
<b>interval [ (p) ]</b>	N/A	12 bytes	-178000000 years to 178000000 years	Time span
<b>json</b>	N/A	1 byte + JSON size	JSON string	Unlimited variable length
<b>lseg</b>	N/A	32 bytes	((x1,y1),(x2,y2))	A line segment on a plane, not allowed in distribution key columns
<b>macaddr</b>	N/A	6 bytes	N/A	Media Access Control (MAC) addresses
<b>money</b>	N/A	8 bytes	-92233720368547758.08 to +92233720368547758.07	Currency amount
<b>path</b>	N/A	16+16n bytes	[(x1,y1),...]	A geometric path on a plane, not allowed in distribution key columns
<b>point</b>	N/A	16 bytes	(x,y)	A geometric point on a plane, not allowed in distribution key columns

Data type	Alias	Storage	Range	Description
<b>polygon</b>	N/A	40+16n bytes	((x1,y1),...)	A closed geometric path on a plane, not allowed in distribution key columns
<b>real</b>	<b>float4</b>	4 bytes	6 decimal digits of precision	Variable precision, inexact
<b>serial</b>	<b>serial4</b>	4 bytes	1 to 2147483647	Auto-increment integer
<b>smallint</b>	<b>int2</b>	2 bytes	-32768 to +32767	Small-range integer
<b>text</b>	N/A	1 byte + string size	Variable-length string	Unlimited variable length
<b>time [ (p) ] [ without time zone ]</b>	N/A	8 bytes	00:00:00[.000000] to 24:00:00[.000000]	Time of day ( without time zone)
<b>time [ (p) ] with time zone</b>	<b>timetz</b>	12 bytes	00:00:00+1359 to 24:00:00-1359	Time of day ( with time zone )
<b>timestamp [ ( p ) ] [ without time zone ]</b>	N/A	8 bytes	4713 BC to 294 ,277 AD	Date and time
<b>timestamp [ ( p ) ] with time zone</b>	<b>timestamptz</b>	8 bytes	4713 BC to 294 ,277 AD	Date and time (with time zone)
<b>xml</b>	N/A	1 byte + XML size	Variable-length XML string	Unlimited variable length

- For more information about the supported standard SQL syntax, see [SQL syntax](#).

## 13 Data Transmission Service (DTS)

---

### 13.1 What is DTS?

**Data Transmission Service (DTS) is a data service provided by Alibaba Cloud. It supports data exchanges between databases of various types such as relational databases and big data systems. DTS supports multiple data transmission methods such as data migration, real-time data subscription, and real-time data synchronization. These data transmission methods are used to achieve data migration with zero downtime, geo-disaster recovery, cache updates, online and offline real-time data synchronization, and asynchronous notifications.**

**DTS applies to multiple business scenarios. It enables you to build a secure, scalable, and highly available architecture.**

### 13.2 Benefits

**DTS supports data transmission between data sources such as relational databases and OLAP databases. It provides multiple data transmission methods such as data migration, change tracking, and data synchronization. Compared with other data migration and synchronization tools, DTS provides better transmission channels because it has high compatibility, high performance, security, and reliability. DTS also provides a variety of features to help you easily create and manage transmission channels.**

#### High compatibility

**DTS supports data migration and synchronization between homogeneous and heterogeneous data sources. For migration between heterogeneous data sources, DTS supports schema conversion.**

**DTS supports multiple data transmission methods including data migration, change tracking, and data synchronization. In change tracking and data synchronization, data is transmitted in real time.**

**Data migration enables you to migrate data between databases with little downtime, which minimizes the impact of data migration on applications. The application downtime during data migration is minimized to several seconds.**

#### High performance

**DTS uses high-end servers to ensure the performance of each data synchronization or migration channel.**

**DTS uses a variety of transmission optimization measures for data migration.**

**Compared with traditional data synchronization tools, the real-time synchronization function of DTS refines the granularity of concurrency to the transaction level. This feature allows you to synchronize the incremental data in one table with multiple concurrent channels, improving synchronization performance.**

#### Security and reliability

**DTS is implemented based on clusters. If a node in a cluster is down or faulty, the control center quickly moves all tasks on this node to another node in the cluster.**

**Secure transmission protocols and tokens are used for authentication across DTS modules to ensure reliable data transmission.**

#### Ease of use

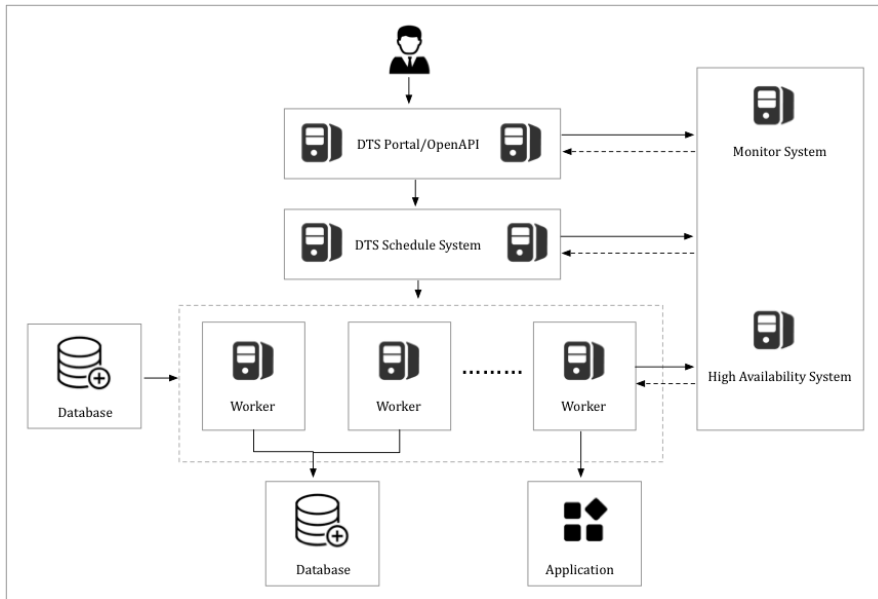
**The DTS console provides a codeless wizard for creating and managing channels.**

**To facilitate channel management, the DTS console displays information about transmission channels, such as transmission status, progress, and performance.**

**DTS supports resumable transmission, and regularly monitors channel status. If DTS detects an error such as network failure or system exception, it automatically fixes the error and restarts the channel. If the error persists, you need to manually repair the channel and restart it in the DTS console.**

## 13.3 Architecture

**The following figure shows the system architecture of DTS.**



## 13.4 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



**Notice:**

- 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.

## 13.5 Features

### 13.5.1 Data migration

#### 13.5.1.1 Data migration

Data migration allows you to quickly migrate data between multiple data sources. Typical scenarios include data migration to the cloud, data migration between instances within Alibaba Cloud, and database split and scale-out. DTS supports data migration between homogeneous and heterogeneous data sources. It also supports ETL features such as data mapping at database, table, and column levels and data filtering.

#### 13.5.1.2 Data sources

DTS supports migrating data between the following data sources.

Table 13-1: Data migration between different data sources

Data source	Schema migration	Full data migration	Incremental data migration
MySQL > RDS for MySQL	Supported	Supported	Supported

Data source	Schema migration	Full data migration	Incremental data migration
MySQL > Oracle	Not supported	Supported	Supported
Oracle > RDS for MySQL	Supported	Supported	Supported
Oracle > RDS for PPAS	Supported	Supported	Supported

### 13.5.1.3 Online migration

Data migration in DTS refers to online data migration that is an automatic process. You need only to specify the source instance, destination instance, and objects for migration. Online data migration supports migration with zero downtime. You must make sure that the DTS server has connections to the source and destination instances at the same time.

### 13.5.1.4 Migration modes

Data migration supports schema migration, full migration, and incremental migration. Descriptions of these migration modes are as follows:

- **Schema migration:** migrates the schema definitions from the source instance to the destination instance.
- **Full migration:** migrates historical data from the source instance to the destination instance.
- **Incremental migration:** migrates incremental data generated during migration from the source instance to the destination instance in real time. You can combine these modes to perform business migration with zero downtime.

### 13.5.1.5 ETL features

Data migration supports the following ETL features:

- **Object name mappings** at database, table, and column levels. Object name mappings are used for data migration between objects on the source and destination instances. The objects have different database, table, or column names.
- **Data filtering** is supported for migration. You can configure a standard SQL filtering criteria to filter the table to be migrated. For example, you can specify the time range to migrate the latest data only.

### 13.5.1.6 Migration task

Migration task is the basic unit of data migration. To migrate data, you must first create a data migration task in the DTS console. To create a data migration task, you must configure information such as the source instance connection type, destination instance connection type, migration type, and objects you want to transfer. You can create, manage, stop, and delete data migration tasks in the DTS console.

## 13.5.2 Data synchronization

### 13.5.2.1 Overview

Real-time data synchronization enables you to synchronize data between two data sources in real time. This feature applies to multiple scenarios, such as zone-disaster recovery, geo-disaster recovery, and data synchronization between OLTP and OLAP databases.

The following table describes synchronization features.

Table 13-2: Synchronization features

Source instance	Destination instance	One-way/two-way synchronization
MySQL	MySQL	<ul style="list-style-type: none"><li>One-way synchronization</li><li>Two-way synchronization</li></ul>
MySQL	MaxCompute	One-way synchronization
MySQL	DataHub	One-way synchronization

### 13.5.2.2 Synchronization tasks

Synchronization tasks are the basic units for real-time data synchronization. To synchronize data between two instances, you must create a synchronization task in the DTS console.

[Table 13-3: Synchronization task statuses and descriptions](#) shows the statuses of a synchronization task during creation and running.



Table 13-3: Synchronization task statuses and descriptions

Status	Description	Available operation
<b>Pre-checking</b>	The synchronization task is performing a pre-check before the task is started.	<ul style="list-style-type: none"> <li>• View synchronization configurations.</li> <li>• Delete the synchronization task.</li> <li>• Replicate synchronization configurations.</li> <li>• Configure monitors and alarms.</li> </ul>
<b>Pre-check failed</b>	The synchronization task has failed the pre-check.	<ul style="list-style-type: none"> <li>• Perform the pre-check.</li> <li>• View synchronization configurations.</li> <li>• Modify synchronization objects.</li> <li>• Modify synchronization speed.</li> <li>• Delete the synchronization task.</li> <li>• Replicate synchronization configurations.</li> <li>• Configure monitors and alarms.</li> </ul>
<b>Not started</b>	The synchronization task that has passed the pre-check is not started.	<ul style="list-style-type: none"> <li>• Perform the pre-check.</li> <li>• Start the synchronization task.</li> <li>• Modify synchronization objects.</li> <li>• Modify synchronization speed.</li> <li>• Delete the synchronization task.</li> <li>• Replicate synchronization configurations.</li> <li>• Configure monitors and alarms.</li> </ul>

Status	Description	Available operation
<b>Initializing</b>	The synchronization task is being initialized.	<ul style="list-style-type: none"> <li>• View synchronization configurations.</li> <li>• Delete the synchronization task.</li> <li>• Replicate synchronization configurations.</li> <li>• Configure monitors and alarms.</li> </ul>
<b>Initialization failed</b>	Data migration has failed during the synchronization task initialization.	<ul style="list-style-type: none"> <li>• View synchronization configurations.</li> <li>• Modify synchronization objects.</li> <li>• Modify synchronization speed.</li> <li>• Delete the synchronization task.</li> <li>• Replicate synchronization configurations.</li> <li>• Configure monitors and alarms.</li> </ul>
<b>Synchronizing</b>	The task is synchronizing data.	<ul style="list-style-type: none"> <li>• View synchronization configurations.</li> <li>• Modify synchronization objects.</li> <li>• Modify synchronization speed.</li> <li>• Pause the synchronization task.</li> <li>• Delete the synchronization task.</li> <li>• Replicate synchronization configurations.</li> <li>• Configure monitors and alarms.</li> </ul>

Status	Description	Available operation
<b>Synchronization failed</b>	<b>A synchronization exception occurred.</b>	<ul style="list-style-type: none"> <li>• <b>View synchronization configurations.</b></li> <li>• <b>Modify synchronization objects.</b></li> <li>• <b>Modify synchronization speed.</b></li> <li>• <b>Start the synchronization task.</b></li> <li>• <b>Delete the synchronization task.</b></li> <li>• <b>Replicate synchronization configurations.</b></li> <li>• <b>Configure monitors and alarms.</b></li> </ul>
<b>Paused</b>	<b>The synchronization task is paused.</b>	<ul style="list-style-type: none"> <li>• <b>View synchronization configurations.</b></li> <li>• <b>Modify synchronization objects.</b></li> <li>• <b>Modify synchronization speed.</b></li> <li>• <b>Start the synchronization task.</b></li> <li>• <b>Delete the synchronization task.</b></li> <li>• <b>Replicate synchronization configurations.</b></li> <li>• <b>Configure monitors and alarms.</b></li> </ul>

### 13.5.2.3 Synchronization objects

- Data synchronization objects include databases, tables, and columns. You can specify the tables that you want to synchronize.
- Data synchronization supports the mapping of database, table, and column names. In other words, objects can have different databases, tables, and column names during data synchronization.
- You can also synchronize specified columns of data in a table.

### 13.5.2.4 Advanced features

The following advanced features are used to facilitate data synchronization:

- Dynamically add and remove synchronization objects

You can add and remove synchronization objects during data synchronization.

- Improve the performance query system

Data synchronization provides the synchronization latency and performance trend chart (RPS and traffic). You can use this to easily view the performance of synchronization links.

## 13.5.3 Data subscription

### 13.5.3.1 Real-time data subscription

Real-time data subscription can help you obtain the incremental data of RDS in real time. You can migrate incremental data based on your business requirements, such as cache updates, asynchronous business decoupling, real-time heterogeneous data synchronization, and real-time complex ETL data synchronization.

Real-time data subscription supports RDS for MySQL instances in classic networks and VPCs.

Real-time data subscription supports the following data sources:

- RDS for MySQL

### 13.5.3.2 Subscription channels and objects

#### Subscription channels

Subscription channels are the basic units of incremental data subscription and consumption. To subscribe to RDS incremental data, you must create a subscription channel in the DTS console for the relevant RDS instance. The subscription channel reads RDS incremental data in real time and stores the most recent increments. You can use the SDK provided by DTS to subscribe to and consume the incremental data in the channel. You can create, manage, and delete subscription channels in the DTS console.

A subscription channel can only be subscribed and consumed by one downstream SDK. To subscribe to an RDS instance for multiple downstream SDKs, you must

create an equivalent number of subscription channels. RDS instances subscribed to with these subscription channels share the same instance ID.

*Table 13-4: Subscription channel statuses and descriptions* shows the statuses of a subscription channel during creation and running.

Table 13-4: Subscription channel statuses and descriptions

Status	Description	Available operation
Pre-checking	The subscription channel has completed task configurations and is performing a pre-check.	Delete the subscription channel.
Not started	The migration task has passed the pre-check, but is not started.	<ul style="list-style-type: none"> <li>• Start subscription</li> <li>• Delete the subscription channel.</li> </ul>
Initializing	The subscription channel is being initialized. This process takes about one minute.	Delete the subscription channel.
Normal	The subscription channel is reading incremental data from an RDS instance.	<ul style="list-style-type: none"> <li>• View sample code.</li> <li>• View the subscribed data.</li> <li>• Delete the subscription channel.</li> </ul>
Abnormal	An exception occurs when the subscription channel reads incremental data from an RDS instance.	<ul style="list-style-type: none"> <li>• View sample code.</li> <li>• Delete the subscription channel.</li> </ul>

## Subscription objects

Subscription objects contain databases and tables. You can specify the tables that you want to subscribe to.

**Incremental data is divided into data update and schema update in data subscription. You can select the specific data type when you configure data subscription.**

### **13.5.3.3 Advanced features**

**The following advanced features are used to facilitate data subscription:**

- **Dynamically add and remove subscription objects**

**You can add and remove subscription objects during data subscription.**

- **View the subscribed data online**

**You can view the incremental data that has been subscribed to in the DTS console.**

- **Modify data consumption time**

**You can modify the time for data consumption at any time.**

## 14 Data Management Service (DMS)

---

### 14.1 What is Data Management Service?

**Data Management Service (DMS) provides centralized management of relational databases and OLAP databases. It is built on the iDB database service platform of Alibaba, and has been providing database development support for tens of thousands of R&D engineers since it was brought online eight years ago. You can use DMS to build your own database DevOps, which improves database R&D efficiency through better self-service and ensures secure employee database access and high database performance.**

**DMS is used to manage relational databases such as MySQL, SQL Server, and PostgreSQL, as well as OLAP databases such as AnalyticDB. It integrates data management with schema management.**

#### 14.1.1 Product value

**DMS provides you with a convenient and secure database access and management platform. Visualized data services enable you to use databases on browsers, eliminating the need to install various database clients. When you edit data online, you can easily perform operations on table data and change table structures, without having to write complex SQL statements. DMS provides advanced functions that common clients do not offer, such as table structure synchronization, database clone, chart-based presentation of result sets, and real-time monitoring.**

**To use DMS, you must first log on to the Apsara Stack console, and then use your database account and password to log on to the DMS console. This feature adds an extra layer of security to your database account. DMS supports HTTPS and SSL for data transmission, and prevents data from being intercepted or tampered with during transmission.**

**DMS also supports RAM and STS for permission verification to prevent unauthorized actions.**

**DMS supports VPC instance access and provides data access interfaces for users while ensuring security of the database instance network, which is beyond the capability of common clients.**

**DMS provides the following benefits:**

- **Simple data operations**
  - **Pain point:** You need a convenient and all-in-one product to complete SQL operations, save common operations, and apply common operations to specific services.
  - **Solution:** You can create a table in DMS and perform operations on table data just as you would in an Excel worksheet. You can add, delete, change, query, and make statistical analysis of table data without understanding SQL. You can customize SQL operations and save common business-related SQL operations in DMS. Then you can apply these operations directly when managing other databases or instances.
- **Visualization of database table structures**
  - **Pain point:** When you design a new business table or perform operations on an existing business table, you often need to understand the structures of all tables in a database. You can execute SQL commands one by one to display the table structures, but this method is neither intuitive nor convenient.
  - **Solution:** Through the document generation function of DMS, you can generate the table structures of an entire database with a single click. Then you can browse these structures online or export them to other formats such as Word, Excel, and PDF.
- **Real-time optimization of database performance**
  - **Pain point:** Detailed monitoring logs over a long period of time are required for database performance optimization. You need to make a detailed analysis of the logs and locate exceptions to better improve the database performance.
  - **Solution:** DMS provides second-level monitoring of database performance metrics, such as SELECT, INSERT, UPDATE, and DELETE operations, the number of active connections, and network traffic volume, and helps keep you informed of any performance variations. DMS allows you to view and terminate database sessions.



- **Chart-based presentation of SQL result sets**

- **Pain point:** Users used to use SQL statements to find data, and import the data into Excel to create static charts such as line charts and pie charts. This process takes a lot of time.
- **Solution:** With DMS, you can directly create charts from SQL result sets. You can also create many advanced charts, such as dynamic charts, period-over-period comparison charts, and personalized tooltips. This helps you produce high-quality work.

- **SQL statement reuse**

- **Pain point:** When you access a database, there is always a need to execute SQL statements. Simple queries are easy to master, while complex analytical queries or queries with certain business logics are not. The cost of rewriting SQL statements each time is too high, and even if the statements are saved to text files, they require constant maintenance and cannot be used flexibly.
- **Solution:** You can use the My SQL function provided by DMS to save frequently used SQL statements. As the SQL statements are not saved locally, they can be reused in any databases or instances.

- **Monitoring of changes to the table data volume**

Big data is the latest trend in data analysis and is widely discussed. However, taking full advantage of the values provided by big data analysis is not an easy task. The core idea of DMS is to start analyzing data when data is available.

DMS monitors changes to table data volume through a custom RDS kernel, which allows it to quickly collect row count changes of each instance, database, and table. DMS provides real-time monitoring, data trends, and detailed data through professional data analysis and interaction.

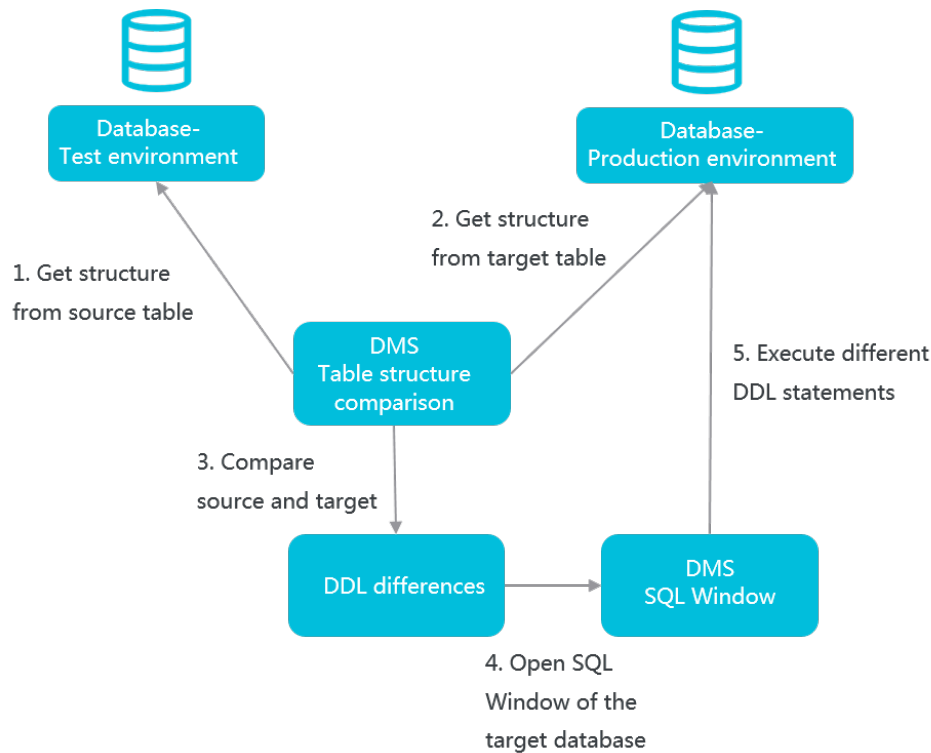
- **Table structure synchronization**

- **Pain point:** Within enterprises, database environments are divided into production environment and test environment. A database will be released in the production environment after it is verified in the test environment. If table structures in the test environment are not completely synchronized to the production environment, major faults can occur during the release.
- **Solution:** You can use the table structural comparison function of DMS to detect inconsistencies in database table structures between the production

and test environments. You can also obtain a DDL statement for table structure correction to ensure table structure consistency between the production and test environments.

*Figure 14-1: DMS - Table structure comparison - Table structure synchronization* shows how to use the table structure comparison function to synchronize table structures.

Figure 14-1: DMS - Table structure comparison - Table structure synchronization



## 14.2 Benefits

### Improved R&D efficiency

- Table schema comparison
- Intelligent SQL prompts
- Convenient reuse of custom SQL statements and SQL templates
- Automatic recovery of working environment
- Export of dictionary files

### Real-time optimization of database performance

- Effective session management
- Monitoring of core metrics in seconds

- Graphical lock management
- Real-time SQL index recommendations
- Reports on overall performance

#### **Comprehensive access security protection**

- Four-layer authentication system
- Fine-grained authorization
- Logon and operation audit

#### **Extensive options for data sources**

- Relational databases such as MySQL, SQL Server, PostgreSQL, and PPAS
- NoSQL databases such as Redis, MongoDB, and Memcache

## **14.3 Architecture**

Apsara Stack Data Management Service (DMS) consists of the business layer, scheduling layer, and connection layer. It processes real-time data access and schedules data-related backend tasks for relational databases.

#### **Business layer**

- The DMS business layer provides online GUI-based database operations, and can be extended linearly to improve the general service capabilities of DMS.
- DMS supports stateless failover, ensuring 24/7 availability.

#### **Scheduling layer**

- The scheduling layer allows you to import and export tables and compare table schemas. This layer schedules tasks by using the thread pool in two modes: real-time scheduling and background periodic scheduling.
- Real-time scheduling allows you to quickly schedule and execute a task on the frontend. After you submit a task, the DMS backend automatically executes the task. After the task is completed, you can download or view the execution result.
- Background periodic scheduling allows you to periodically obtain specified data, such as data trends. DMS collects business data in the background based on scheduled tasks for your reference and analysis.

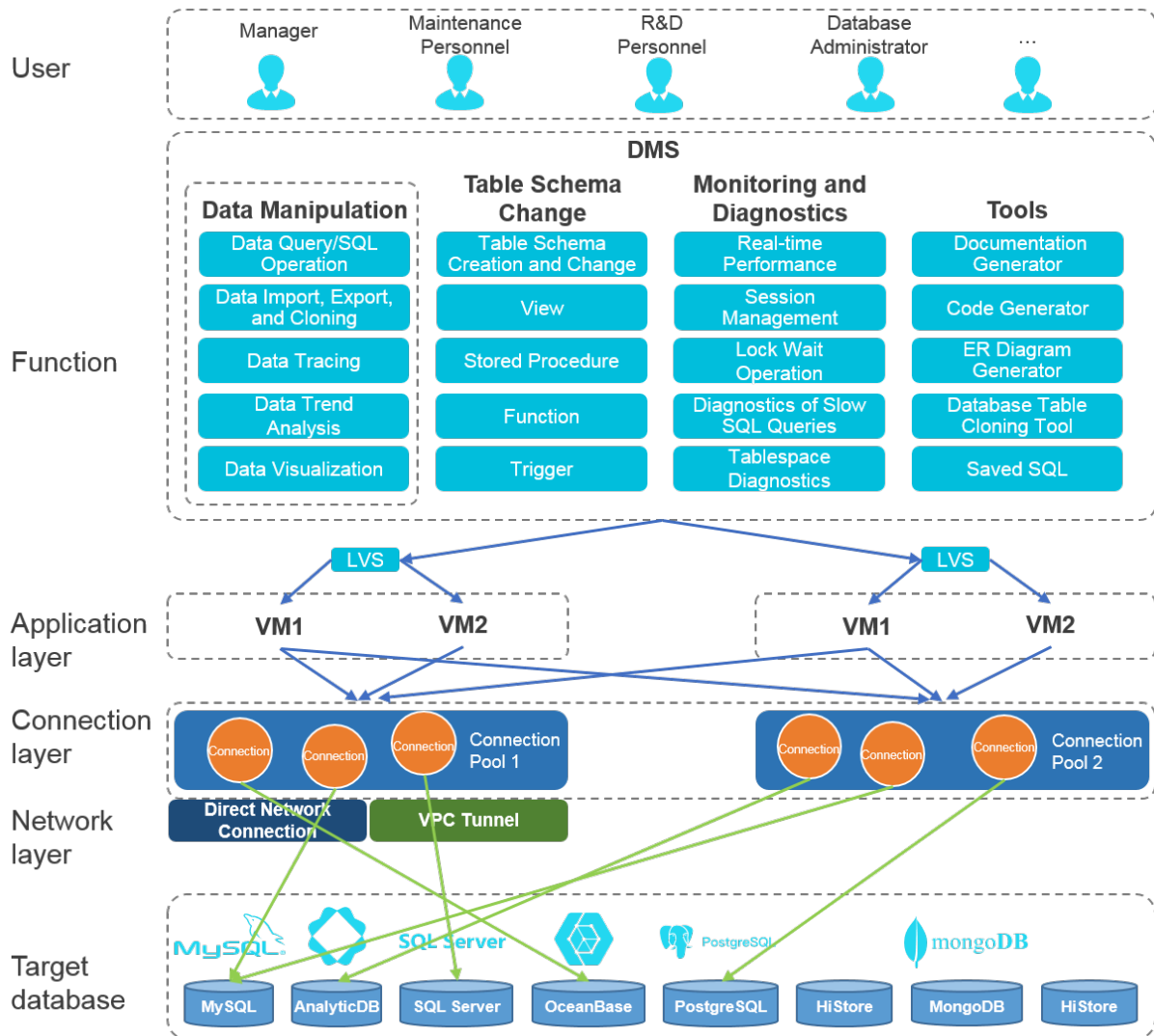
## Connection layer

**The connection layer is the core component for data access in DMS. It has the following features:**

- **Processes requests from MySQL, SQL Server, PostgreSQL, and AnalyticDB databases.**
- **Supports session isolation and persistence. SQL windows opened in DMS are isolated from each other and the sessions in each SQL window are persistent to simulate the client experience.**
- **Controls the number of instance sessions to avoid establishing a large number of connections to a single instance.**

- Provides different connection release policies for different functions. This improves user experience and reduces the number of connections to the databases.

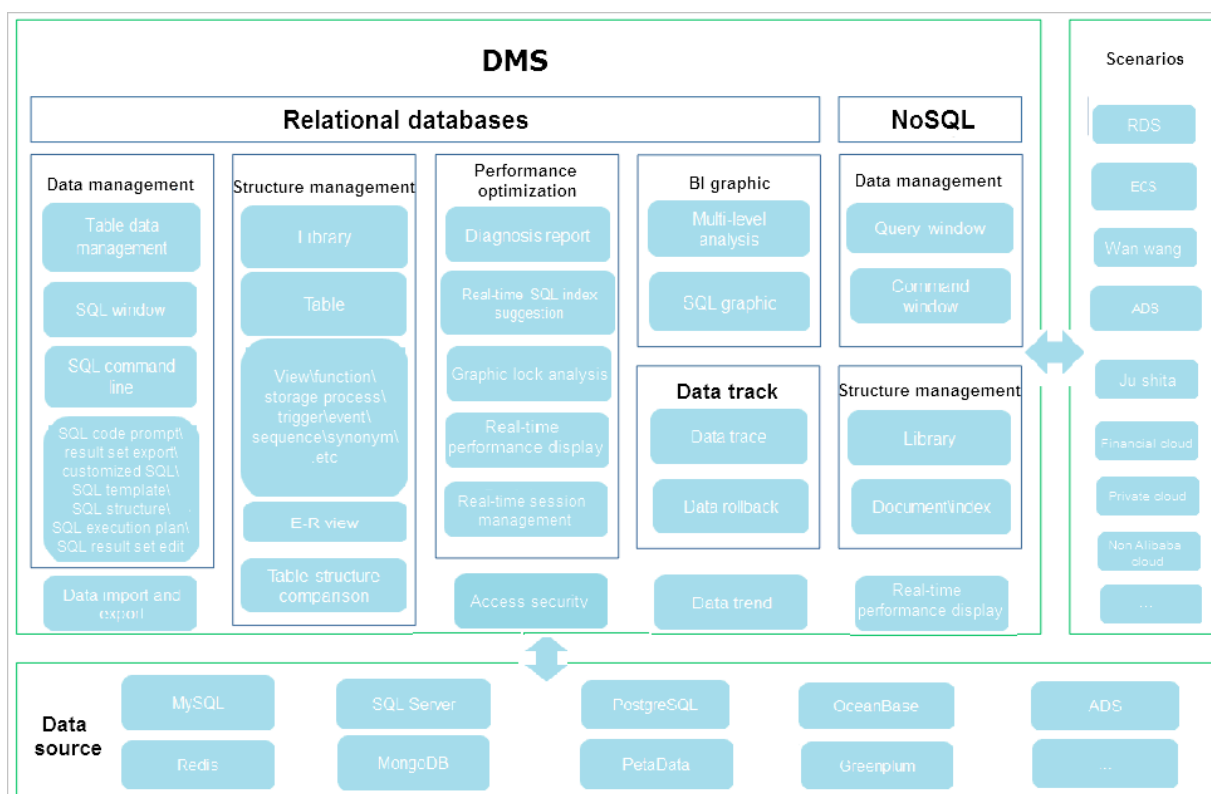
Figure 14-2: DMS system architecture



## 14.4 Features

The following figure shows the features of DMS.

Figure 14-3: DMS features



- **Features for relational databases**
  - **Data management:** SQL editor, SQL command-line interface, table management, smart SQL completion, SQL formatting, custom SQL queries, SQL templates, SQL execution plans, and import and export.
  - **Schema management:** schema comparison and management for objects such as databases, tables, views, functions, storage procedures, triggers, events, series, and synonyms.
  - **Performance optimization:** real-time performance monitoring, real-time SQL index recommendation, graphical interface for lock management, session management, and diagnostic reporting.
  - **Access control:** four-layer authentication, logon and operation auditing, and fine-grained authorization at the Apsara Stack tenant account, access address, and feature levels.

- **Features for NoSQL databases**
  - **Data management: query editor and command-line interface.**
  - **Schema management: management of objects such as databases, documents, and indexes.**
  - **Real-time performance monitoring: real-time display of key performance indicators.**

## 15 Server Load Balancer (SLB)

---

### 15.1 What is Server Load Balancer?

**Server Load Balancer (SLB) is a type of traffic control service that distributes inbound traffic across multiple ECS instances based on forwarding rules. SLB improves the service capability and availability of applications.**

**SLB consists of three components:**

- **SLB instances:** An SLB instance is a running load-balancing service that receives and distributes inbound traffic to back-end servers.

**To use the SLB service, you must create an SLB instance with at least one listener and two ECS instances configured.**

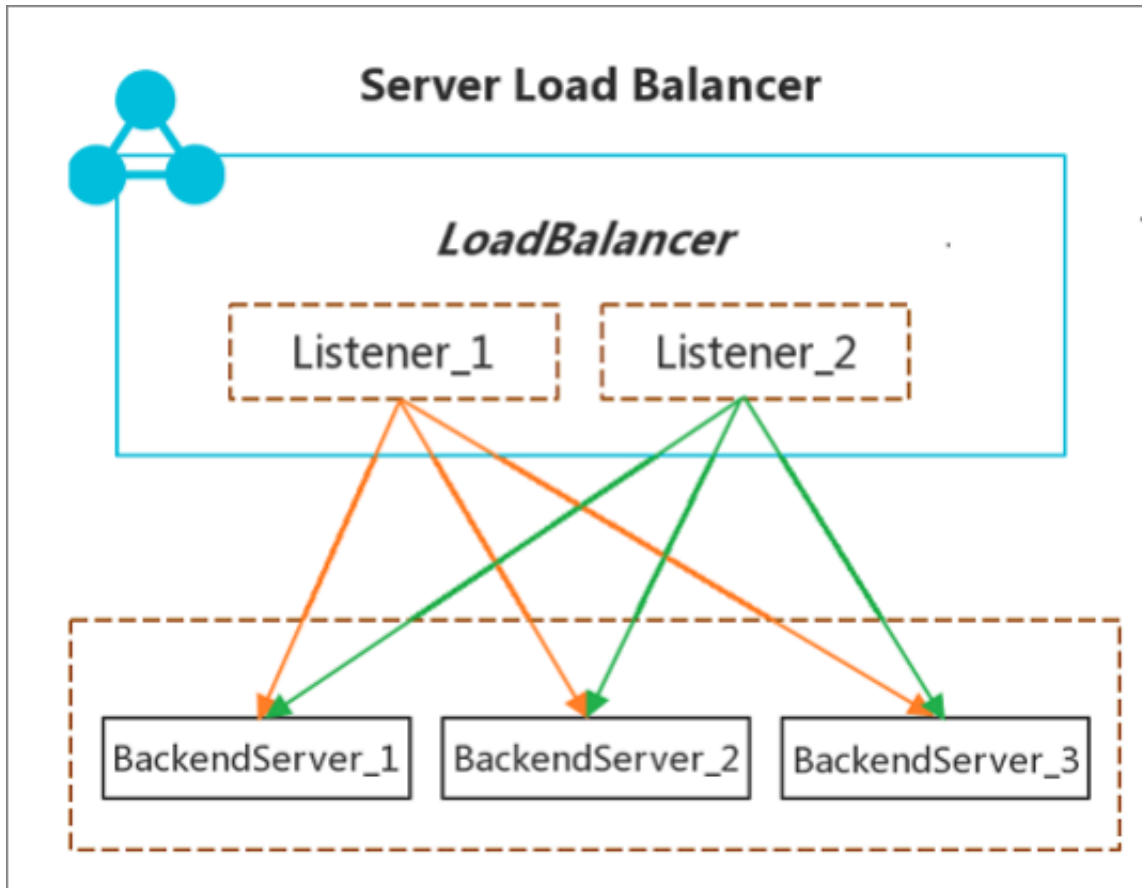
- **Listeners:** A listener checks client requests and forwards them to back-end servers. It also performs health check on the back-end servers.

**You can create Layer-4 (TPC/UDP) or Layer-7 (HTTP/HTTPS) listeners as needed. You can create domain- and URL- based forwarding rules for Layer-7 listeners.**

- **Backend servers:** Backend servers are ECS instances attached to an SLB instance to receive and process the distributed requests. You can divide ECS instances running different applications or functioning different roles into different server groups.

**As shown in the following figure, after the SLB instance receives a client request , the listener forwards the request to the corresponding back-end ECS instances based on the configured forwarding rules.**





## 15.2 Architecture

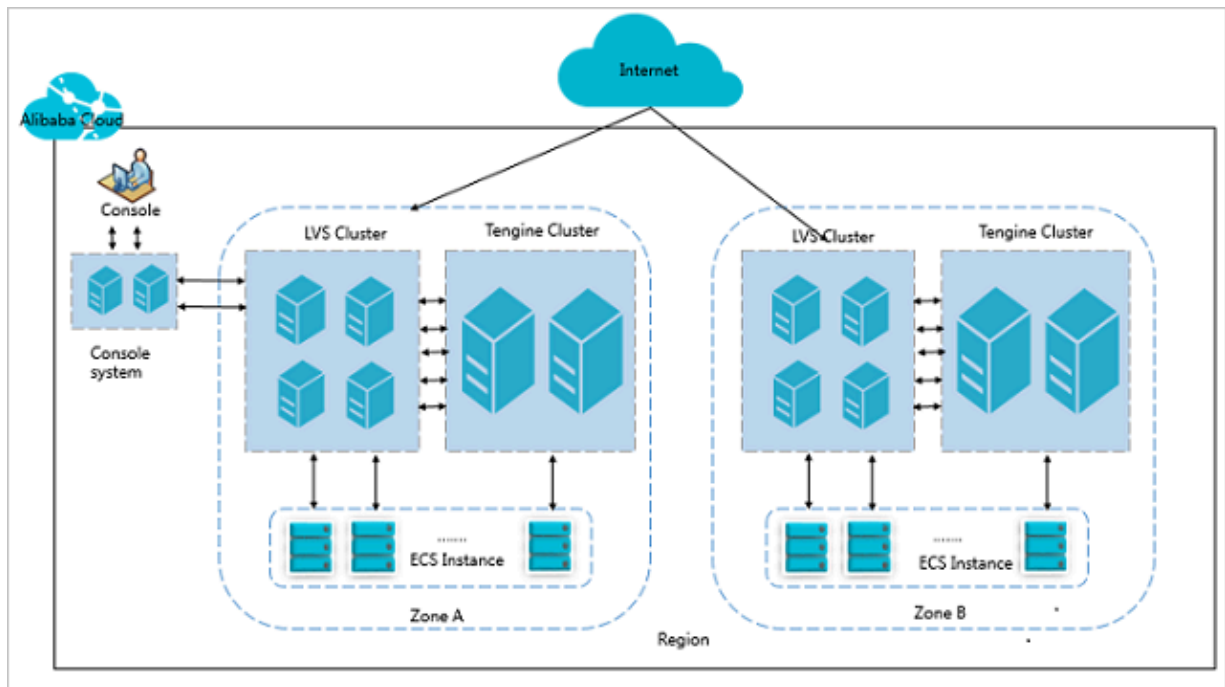
SLB is deployed in clusters to achieve session synchronization. This can eliminate SPOFs of back-end servers, improve redundancy, and ensure service stability.

Apsara Stack provides Layer-4 (TCP and UDP) and Layer-7 (HTTP and HTTPS) load-balancing services.

- Layer-4 SLB combines the open-source Linux Virtual Server (LVS) with Keepalived to balance loads, and implements customized optimizations to meet cloud computing requirements.

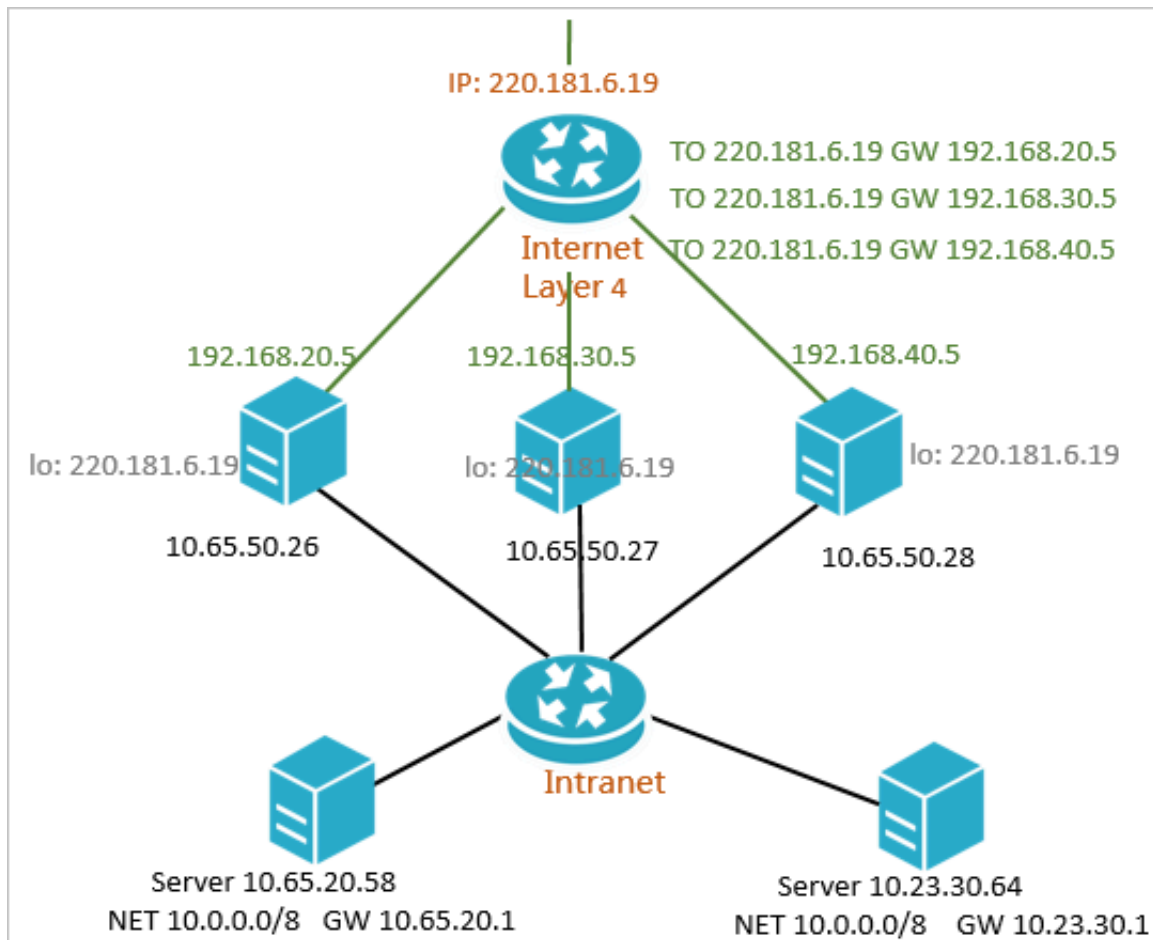
- Layer-7 SLB uses Tengine to balance loads. Tengine is a Web server project launched by Taobao. Based on NGINX, Tengine has a wide range of advanced features enabled for high-traffic websites.

Figure 15-1: SLB architecture



As shown in the following figure, Layer-4 SLB runs in a cluster of LVS machines. The cluster deployment model strengthens the availability, stability, and scalability of load-balancing services in abnormal circumstances.

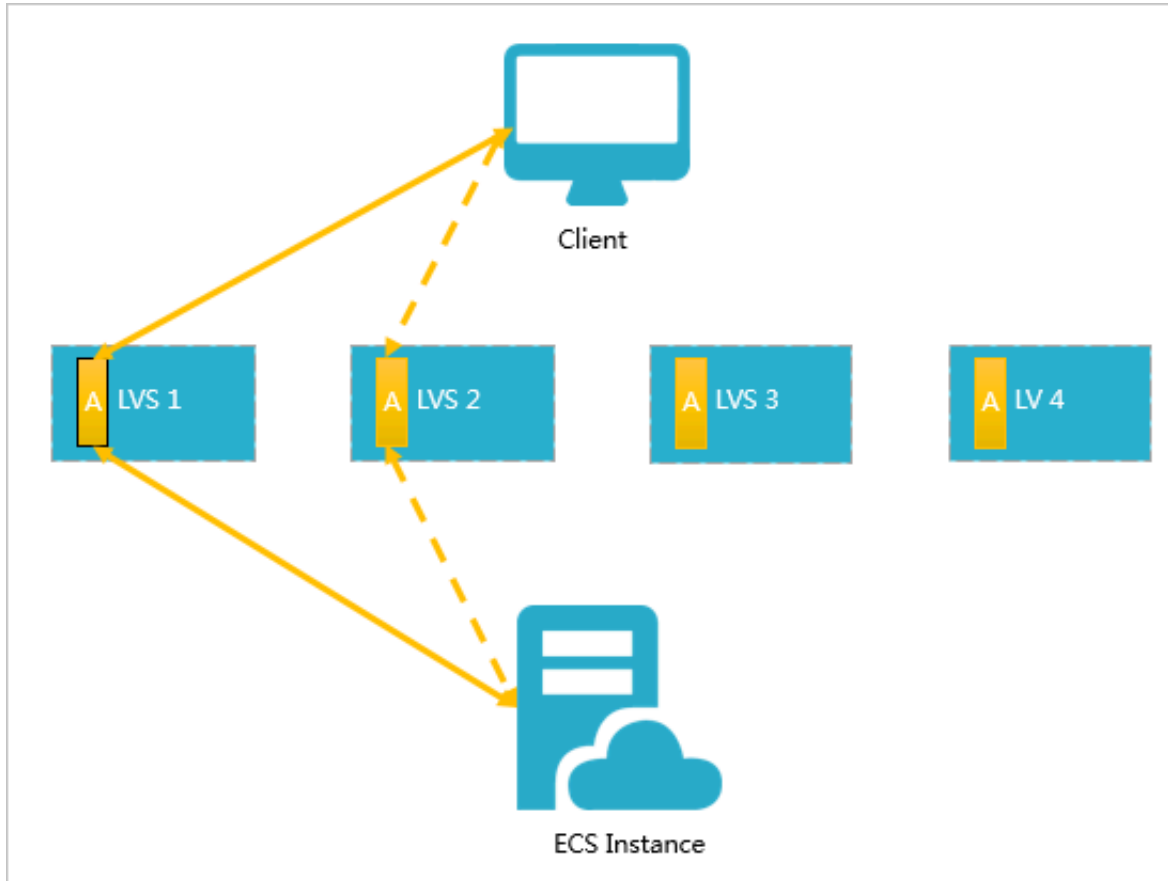
Figure 15-2: Cluster deployment



Each machine in the LVS cluster uses multicast packets to synchronize sessions with the other LVS machines. As shown in the following figure, Session A established on LVS1 is synchronized to other LVS machines after the client transfers three data packets to the server. Solid lines indicate the current active connections, while dotted lines indicate that the session requests will be sent to other normally working machines if LVS1 fails or is being maintained. In this way, you can perform

hot updates, machine failure maintenance, and cluster maintenance without affecting business applications.

Figure 15-3: Session synchronization



## 15.3 Features

Server Load Balancer (SLB) is a type of traffic control service that distributes inbound traffic across multiple ECS instances based on forwarding rules. SLB improves the service capability and availability of applications.

You can use SLB to virtualize multiple ECS instances in the same region into an application server pool with high performance and high availability by configuring virtual IP addresses (VIPs). Then, you can distribute client requests to the ECS instances based on forwarding rules.

SLB checks the health status of the ECS instances and automatically isolates abnormal ones in the server pool to eliminate single points of failure (SPOFs), improving the overall service capability of applications. SLB is also well equipped to defend against DDoS attacks.

## 15.4 Benefits

### 15.4.1 LVS in Layer-4 SLB

**This topic describes the customized technical improvements on the standard LVS performed by Alibaba Cloud.**

Drawbacks of the standard LVS

**Linux Virtual Server (LVS) is the world's most popular Layer-4 load-balancing software. LVS was developed by Dr. Zhang Wensong in May 1998 for Linux systems . LVS is a kernel module based on a linux netfilter framework named IP Virtual Server (IPVS), which is similar to iptables. LVS is hooked into LOCAL\_IN and FORWARD.**

**In a large-scale cloud computing network, the standard LVS has the following drawbacks:**

- **Drawback 1: LVS supports three packet forwarding modes: NAT, DR, and TUNNEL. When these forwarding modes are deployed in a network with multiple VLANs, the network topology becomes complex and incurs high O&M costs.**
- **Drawback 2: Compared with commercial load-balancing devices such as F5, LVS lacks defense against DDoS attacks.**
- **Drawback 3: LVS uses PC servers and the Virtual Router Redundancy Protocol ( VRRP) of Keepalived to deploy primary and secondary nodes for high availability . Therefore, its performance cannot be extended.**
- **Drawback 4: The configurations and health check performance of the Keepalived software are insufficient.**

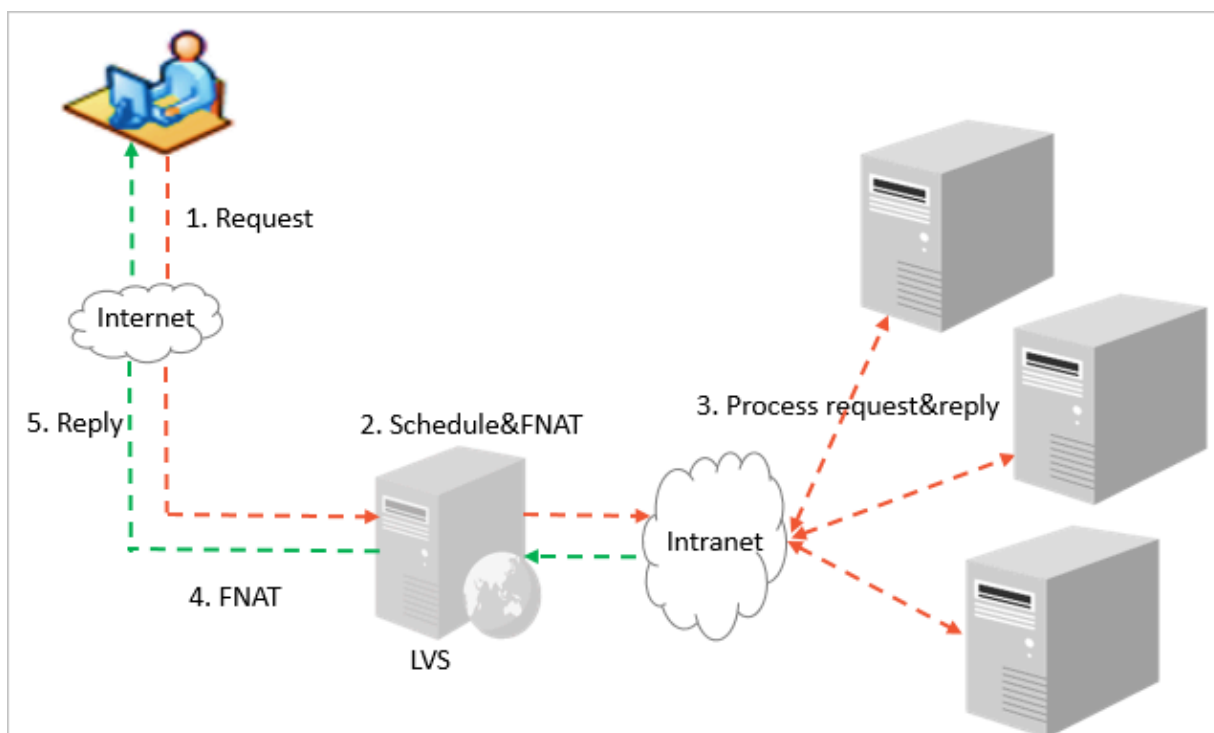
LVS customized features

**To solve these problems, Alibaba Cloud added the following customized features to LVS. For more information about Ali-LVS, visit <https://github.com/alibaba/LVS>.**

- **Customization 1: FULLNAT, a new forwarding mode that enables inter-VLAN communication between LVS load balancers and back-end servers.**
- **Customization 2: Defense modules such as SYNPROXY against TCP flag-targeted DDoS attacks.**
- **Customization 3: Support for LVS cluster deployment.**
- **Customization 4: Improved Keepalived performance.**

## FULLNAT technology

- **Principles:** The module introduces local addresses (internal IP addresses). IPVS translates cip-vip to and from lip-rip, in which both lip and rip are internal IP addresses. This means that the load balancers and back-end servers can communicate across VLANs.
- **All inbound and outbound data flows traverse LVS.** 10-GE Network Interface Cards (NICs) are used to ensure adequate bandwidth.
- **Currently, FULLNAT supports only TCP.**



## SYNPROXY technology

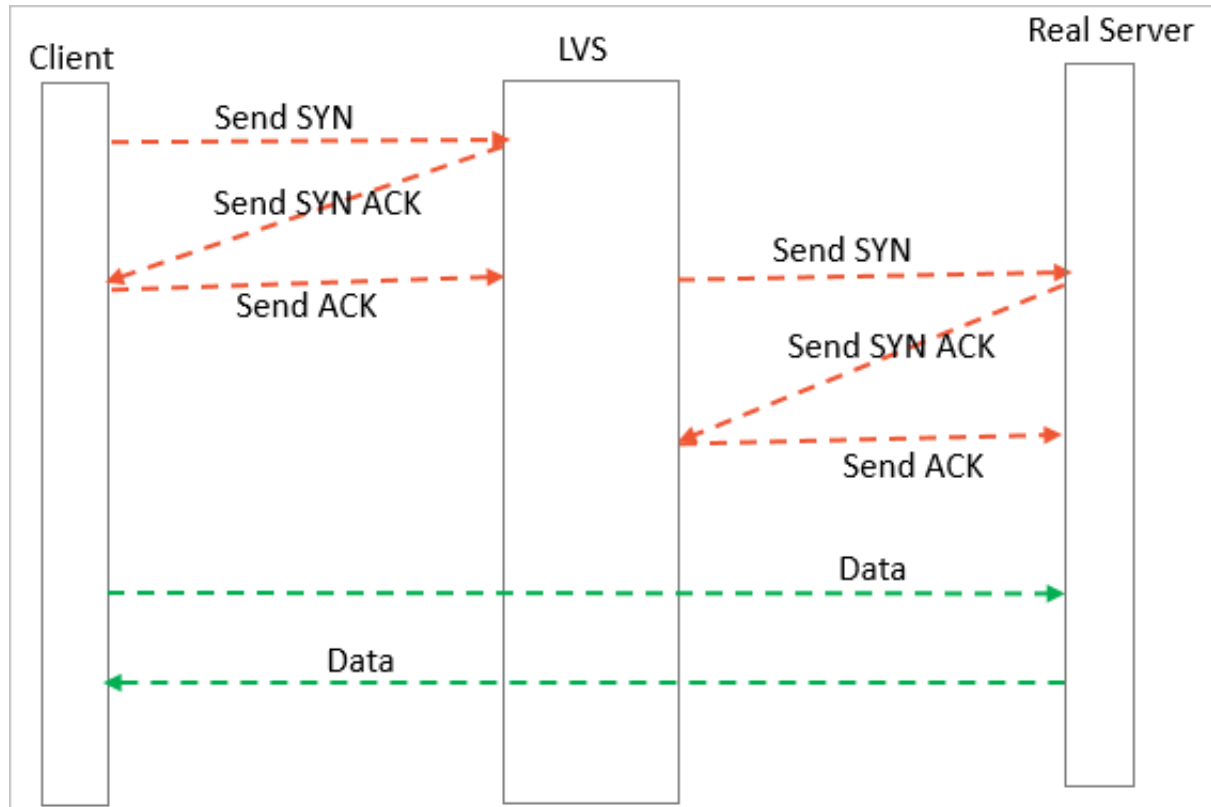
**LVS uses the SYNPROXY module to defend against TCP flag-targeted attacks and Synflood attacks (both are DDoS attacks). According to the principle of SYN cookies in the Linux TCP protocol stack, LVS acts as a proxy for TCP three-way handshakes.**

**The process is as follows:**

1. A client sends an SYN packet to LVS.
2. LVS constructs an SYN-ACK packet with a unique sequence number and sends this packet to the client. The client returns an ACK response to LVS.

3. LVS checks whether the `ack_seq` value in the ACK response is valid. If the value is valid, LVS establishes a three-way handshake with a back-end server.

Figure 15-4: LVS proxy of a three-way handshake



To defend against ACK, FIN, and RST flood attacks, LVS checks the connection table and discards any requests for connections which are undefined in the table.

#### Cluster deployment

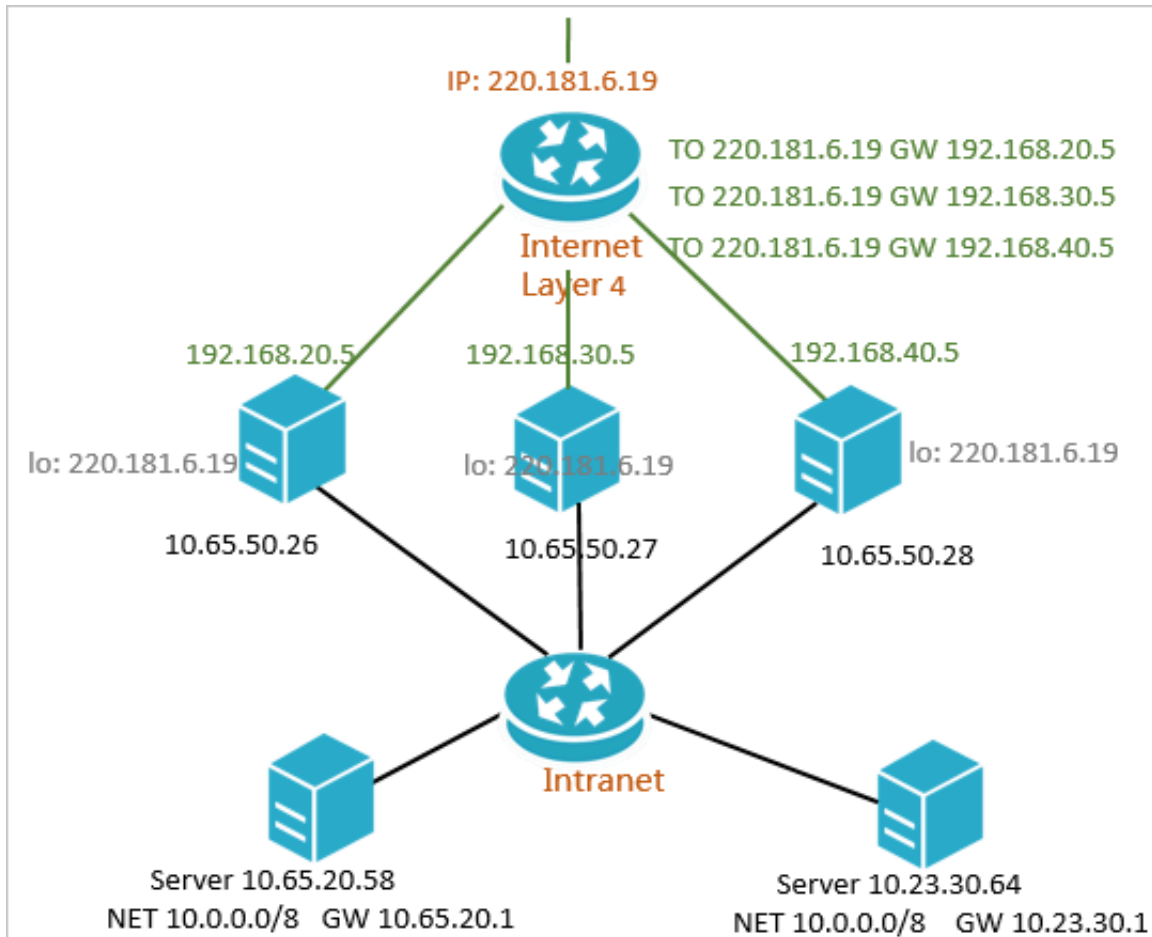
An LVS cluster communicates with uplink switches through OSPF. The uplink switches use equal-cost multi-path (ECMP) routes to distribute traffic to the LVS cluster. Then, the LVS cluster forwards the traffic to your servers.

The cluster deployment model ensures the stability of Layer-4 SLB with the following features:

- **Robustness:** LVS and uplink switches use OSPF as the heartbeat protocol. A virtual IP address (VIP) is configured on all LVS nodes in the cluster. The switches can locate the failure of any LVS node and remove it from the ECMP route list.

- **Scalability:** You can scale out an LVS cluster if traffic from a VIP exceeds the cluster capacity.

Figure 15-5: Cluster deployment



Keepalived optimization

**Improvements made to Keepalived include:**

- Changing the asynchronous network model from select to epoll.
- Optimizing the reload process.

Features of Layer-4 SLB

**In conclusion, Layer-4 SLB has the following features:**

- **High availability:** The LVS cluster ensures redundancy and prevents SPOFs.
- **Security:** Together with Apsara Stack Security, LVS provides near real-time defense.
- **Health check:** LVS performs health check on back-end ECS instances and automatically isolates abnormal instances until they recover.



## 15.4.2 Tengine in Layer-7 SLB

Tengine is a Web server project launched by Alibaba. Based on NGINX, Tengine has a wide range of advanced features enabled for high-traffic websites. NGINX is one of the most popular open-source Layer-7 load-balancing software.

For more information about Tengine, visit <http://tengine.taobao.org/>.

### Customized features

**Tengine is customized for cloud computing scenarios:**

- **Inherits all features of NGINX 1.4.6 and is fully compatible with NGINX configurations.**
- **Supports the dynamic shared object (DSO) module. This means you do not need to recompile Tengine to add a module.**
- **Provides enhanced load balancing capabilities, including a consistent hash module and a session persistence module. It can also actively perform health checks on back-end servers and automatically enable or disable servers based on their status.**
- **Monitors system loads and resource usage to protect the system.**
- **Provides error messages to help locate abnormal servers.**
- **Provides an enhanced protection module (by limiting the access speed).**

### Features of Layer-7 SLB combined with Tengine

**Layer-7 Server Load Balancer (SLB) is based on Tengine, and has the following features:**

- **High availability:** The Tengine cluster ensures redundancy and prevents single points of failure (SPOFs).
- **Security:** Tengine provides multi-dimensional protection against CC attacks.
- **Health check:** Tengine performs health check on back-end ECS instances and automatically isolates abnormal instances until they recover.
- **Supports Layer-7 session persistence.**
- **Supports consistent hash scheduling.**

# 16 Virtual Private Cloud (VPC)

---

## 16.1 What is VPC?

**A Virtual Private Cloud (VPC) is a private network established in Apsara Stack. VPCs are logically isolated from each other.**

### Background information

**The continuous development of cloud computing technologies leads to increasing virtual network requirements such as scalability, security, reliability, privacy, and performance. This scenario has hastened the birth of a variety of network virtualization technologies.**

**Earlier solutions combined virtual and physical networks to form a flat network architecture, such as large layer-2 networks. As the scale of virtual networks grew, earlier solutions faced more serious problems. A few notable problems include ARP spoofing, broadcast storms, and host scanning. Various network isolation technologies emerged to resolve these problems by completely isolating the physical networks from the virtual networks. One of the technologies utilized VLAN to isolate users, but due to VLAN limitations, it could only support up to 4096 nodes. It is insufficient to support the huge amount of users in the cloud.**

### Benefits

**A VPC has the following benefits:**

- **Security**

**Each VPC is identified by a unique tunnel ID. Different VPCs are isolated by tunnel IDs.**

- **Ease of use**

**You can create and manage a VPC in the VPC console. After a VPC is created, the system automatically creates a VRouter and a routing table for it.**

- **Scalability**

**You can create multiple subnets in a VPC to deploy different services. Additionally, you can connect a VPC to a local IDC or another VPC to extend the network architecture.**

## Scenarios

**VPC applies to scenarios with high requirements on communication security and service availability.**

- **Host applications**

**You can host applications that provide external services in a VPC and control access to these applications from the public network by creating security group rules and access control 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 public network, and deploy its application databases in a subnet that cannot access the public network.**

- **Host applications that require public network access**

**You can host an application that requires access to the public network in a subnet of a VPC and route the traffic through NAT. After you configure SNAT rules, instances in the subnet can access the public network without exposing their private IP addresses, which can be changed to public IP addresses anytime to avoid external attacks.**

- **Zone-disaster recovery**

**You can divide a VPC into one or multiple subnets by creating VSwitches. Different VSwitches within the same VPC can communicate with each other. Resources can be deployed to VSwitches of different zones to achieve zone-disaster recovery.**

- **Isolate business systems**

**VPCs are logically isolated from each other. To isolate multiple business systems, such as the production and test environments, you can create a VPC for each environment. When the VPCs need to communicate with each other, you can create a peering connection between them.**

- **Extend the local network architecture**

**To extend the local network architecture, you can connect the local IDC to a VPC. You can also seamlessly migrate local applications to the cloud without changing the application access method.**

## 16.2 Benefits

A VPC is a logically isolated virtual network based on the mainstream tunneling technology.

Each VPC is identified by a unique tunnel ID. Different VPCs are isolated by tunnel IDs:

- Similar to traditional networks, VPCs can also be divided into subnets. ECS instances in the same subnet use the same VSwitch to communicate with each other, whereas ECS instances in different subnets use VRouters to communicate with each other.
- VPCs are completely isolated from each other and can only be interconnected by mapping an external IP address (EIP or NAT IP address).
- The IP packets of an ECS instance are encapsulated by using the tunneling technology. Therefore, information about 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 VPCs use security groups as firewalls to control the traffic to and from ECS instances. This way, ECS instances in different VPCs are isolated at Layer 3.

## 16.3 Architecture

A VPC is a private network logically isolated from other virtual networks.

Network architecture

Each VPC consists of a private Classless Inter-Domain Routing (CIDR) block, a VRouter, and at least a VSwitch.

- CIDR blocks

A CIDR block is a private IP address range in a VPC. The IP addresses of all cloud resources deployed in the VPC are within the specified CIDR block. When

creating a VPC or a VSwitch, you must specify the private IP address range in the form of a CIDR block.

You can use any of the following standard CIDR blocks and their subnets as the IP address range of the VPC.

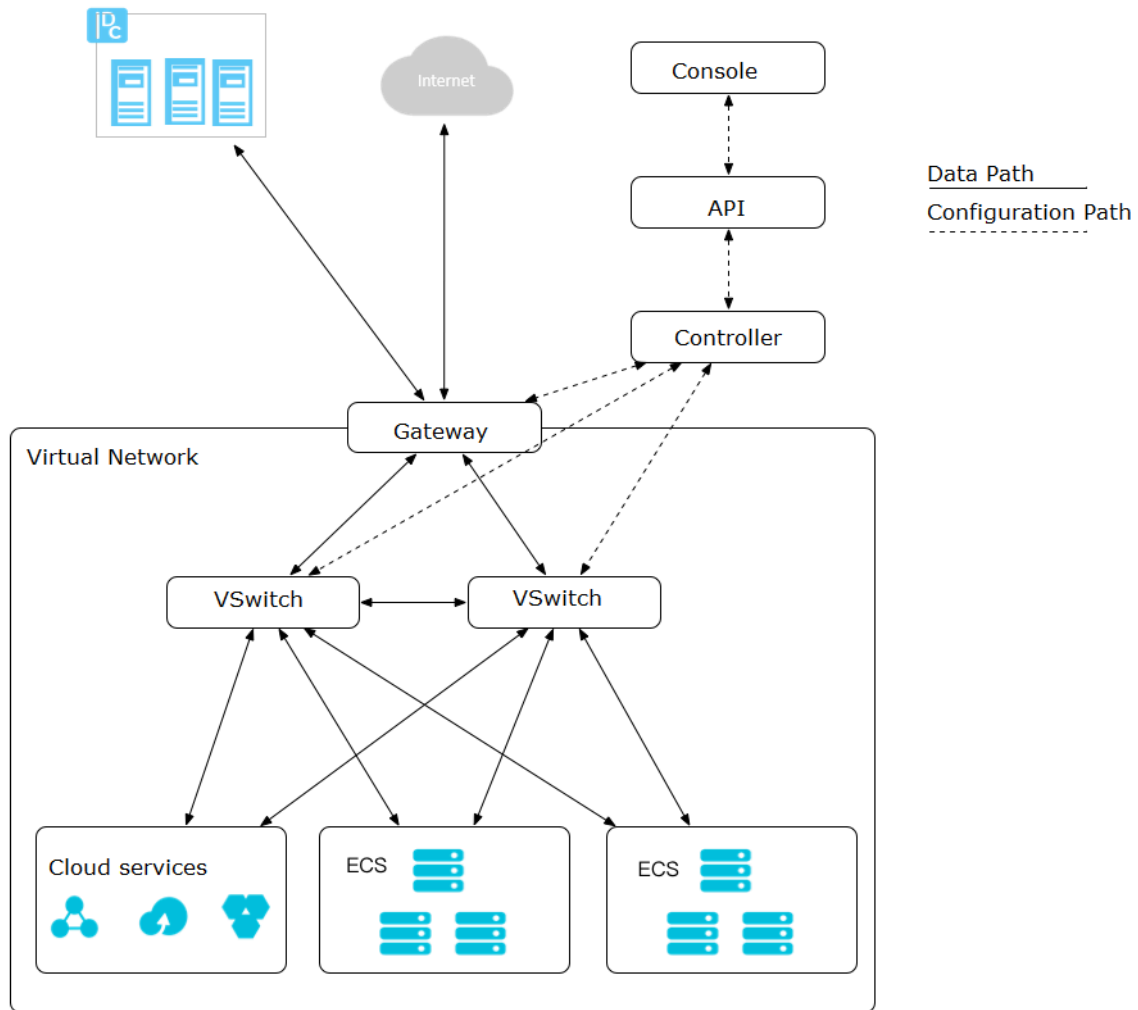
CIDR block	Number of available private IP addresses (system reserved ones excluded)
192.168.0.0/16	65,532
172.16.0.0/12	1,048,572
10.0.0.0/8	16,777,212

- **VRouters**

A VRouter is the hub of a VPC. A VRouter is also an important component of a VPC. The VRouter connects the VSwitches in a VPC and serves as the gateway connecting the VPC with other networks. After you create a VPC, the system automatically creates a VRouter, which is associated with a routing table.

- **Switches**

A VSwitch is a basic network device in a VPC and is used to connect different cloud product instances. After creating a VPC, you can further divide the VPC into one or more subnets by creating VSwitches. The VSwitches within a VPC are interconnected. You can deploy applications in VSwitches of different zones to improve the service availability.

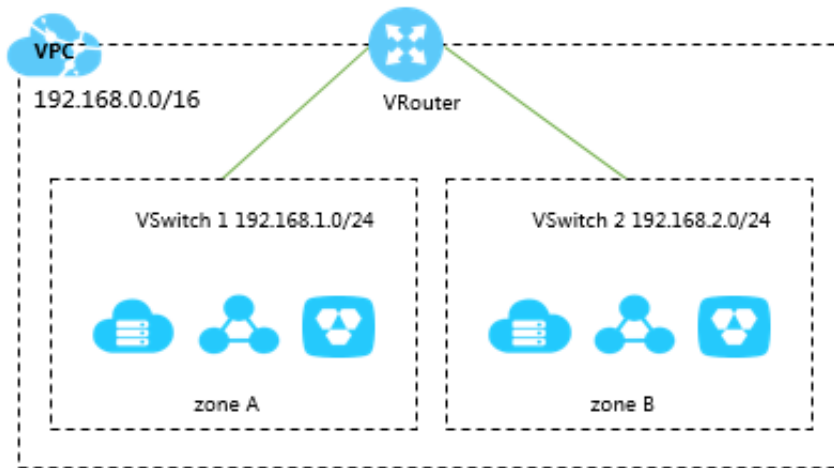


### System architecture

**The VPC architecture contains the VSwitches, gateway, and controller. The VSwitches and gateway form the key data path. Controllers use the protocol developed by Alibaba Cloud to forward the forwarding table to the gateway and VSwitches, completing the key configuration path. In the overall architecture, the configuration path and data path are separated from each other. VSwitches are distributed nodes. The gateway and controller are deployed in clusters. Multiple data centers are built for backup and disaster recovery. Redundant links**

are provided for disaster recovery. This deployment mode improves the overall availability of the VPC.

Figure 16-1: VPC architecture



## 16.4 Features

A VPC is a logically isolated virtual network based on the mainstream tunneling technology.

Each VPC is identified by a unique tunnel ID. A unique tunnel ID is generated when tunnel encapsulation is performed on each data packet transmitted between the ECS instances within a VPC. Then, the data packet is transmitted over the physical network. ECS instances in different VPCs cannot communicate with each other. They have different tunnel IDs and therefore are on different routing planes.

Alibaba Cloud developed technologies such as the VSwitch, Software Defined Network (SDN), and hardware gateway based on the tunneling technology. These technologies serve as the basis for VPCs.

# 17 Log Service

---

## 17.1 What is Log Service?

### 17.1.1 Overview

**Log Service is a unified solution for high volumes of log data, and provides log data collection, subscription, query, and transfer functions.**

- **Real-time collection and consumption:** Log Service collects log data in real time from multiple channels through the client, APIs, tracking.js, and libraries. Data can be immediately subscribed and read after it is written. Interfaces such as Spark Streaming, Storm, and Consumer Library can be used to process data in real time.
- **LogSearch:** LogSearch creates indexes for log data in real time and provides real-time and powerful storage and query engines. LogSearch allows you to retrieve logs by various dimensions such as time, keyword, and context.

**Log Service can automatically scale based on processing requirements. It can scale out to handle large volumes (PBs) of data.**

### 17.1.2 Values

**Log Service helps you build solutions for large volumes of log data.**

**Log Service is applicable to the following scenarios: data collection, real-time computing, data warehousing and offline analysis, product operation and analysis, operations and maintenance, and management.**

- **Data collection and consumption**
- **ETL and stream processing**
- **Event sourcing and tracing**
- **Log management**

## 17.2 Benefits



## 17.2.1 Features

### Real-time log collection (LogHub)

**LogHub implements real-time log collection. LogHub uses various methods to collect data for real-time downstream consumption.**

- **Log collection using Logtail: stable, reliable, and secure. This method provides high performance at low resource consumption. This method is available for all platforms including Linux, Windows, and Docker.**
- **Log collection using APIs or SDKs: flexible, convenient, and scalable. This method is available on mobile terminals and supports multiple programming languages.**
- **Cloud service log collection: LogHub can collect logs from other cloud services, such as ECS, in a convenient and efficient manner.**
- **Other log collection methods: include syslog, Unity3D, Logstash, Log4j, and NGINX.**

### Real-time log consumption (LogHub)

**LogHub supports stream computing, collaborative consumption libraries, and multiple programming languages.**

- **Comprehensive features: LogHub provides all of the features of Kafka. It also has order-preserving, elastic scaling, and time-frame-based seeking features.**
- **Stability and reliability: LogHub supports immediate consumption of written data, multiple data copies, and fast elastic scaling, and achieves low costs.**
- **Easy to use: LogHub supports Spark Streaming, Storm, the Consumer Library automatic load balancing mode, and SDK subscriptions.**

### Log query (LogSearch)

**LogSearch provides real-time log indexing and querying. LogSearch creates indexes for logs and supports log searching by time and keyword.**

- **Large scale: LogSearch supports real-time indexing for PBs of data.**
- **Flexible queries: LogSearch supports keyword-based queries, fuzzy matching, cross-topic queries, and context queries.**

## 17.2.2 Service benefits

### Fully managed service

- **Log Service is easy to access and use.**
- **LogHub has all of the functions of Kafka, provides complete functional data such as monitoring and alert data, and supports elastic scaling (by PB/day).**
- **LogSearch/Analytics provides functions such as quick query, dashboard, and alert.**
- **Log Service supports over 30 access methods, and can be seamlessly connected with open source software (Storm and Spark).**

### Comprehensive ecosystem

- **LogHub supports over 30 types of log data sources such as embedded devices, Web pages, servers, and programs. LogHub can also be interconnected with consumption systems such as Storm and Spark Streaming.**
- **LogSearch/Analytics is compatible with SQL-92, has complete query and analysis syntax, supports JDBC, and can interconnect with Grafana.**

### Strongly real-time

- **LogHub: Data can be used immediately after being written. The Logtail data collection agent collects and transfers data in real time.**
- **LogSearch/Analytics: Data can be queried and analyzed immediately after being written. In the situation where multiple query and analysis conditions are specified, data can be queried and analyzed within seconds.**

### Complete API operations and SDKs

- **Log Service supports user-defined management and secondary development.**
- **All Log Service functions can be implemented by using API operations and SDKs. SDKs in multiple programming languages are provided so that you can easily manage services for millions of devices.**
- **The syntax for query and analysis is simple and compatible with SQL-92. User-friendly interfaces can be used to interconnect with software in the ecosystem.**

## 17.3 Architecture

## 17.3.1 Components

### Logtail

**The Logtail agent collects logs. It has the following characteristics:**

- **Non-intrusive file-based log collection**
  - **Logtail reads only files.**
  - **Log collection is not intrusive.**
- **High security and reliability**
  - **Logtail can rotate files without data loss.**
  - **Logtail supports local caching.**
  - **Logtail is retried when network exceptions occur.**
- **Convenient management**
  - **Logtail can be accessed through a Web client.**
  - **Logtail supports UI visualization.**
- **Comprehensive self-protection**
  - **Logtail monitors the CPU and memory usage of its processes in real time.**
  - **Logtail sets an upper limit on the resource usage of its processes.**

### Front-end servers

**Front-end servers are the front-end machines built on LVS and NGINX. They have the following characteristics:**

- **Support for HTTP and REST**
- **Scale-out**

**Front-end servers can be quickly added when traffic rises to increase processing capabilities.**

- **High throughput, low latency**
  - **Asynchronous processing: If an exception occurs when a single request is sent , other requests are not affected.**
  - **LZ4 compression: The processing capabilities of individual servers are increased while network bandwidth consumption is reduced.**

## Back-end servers

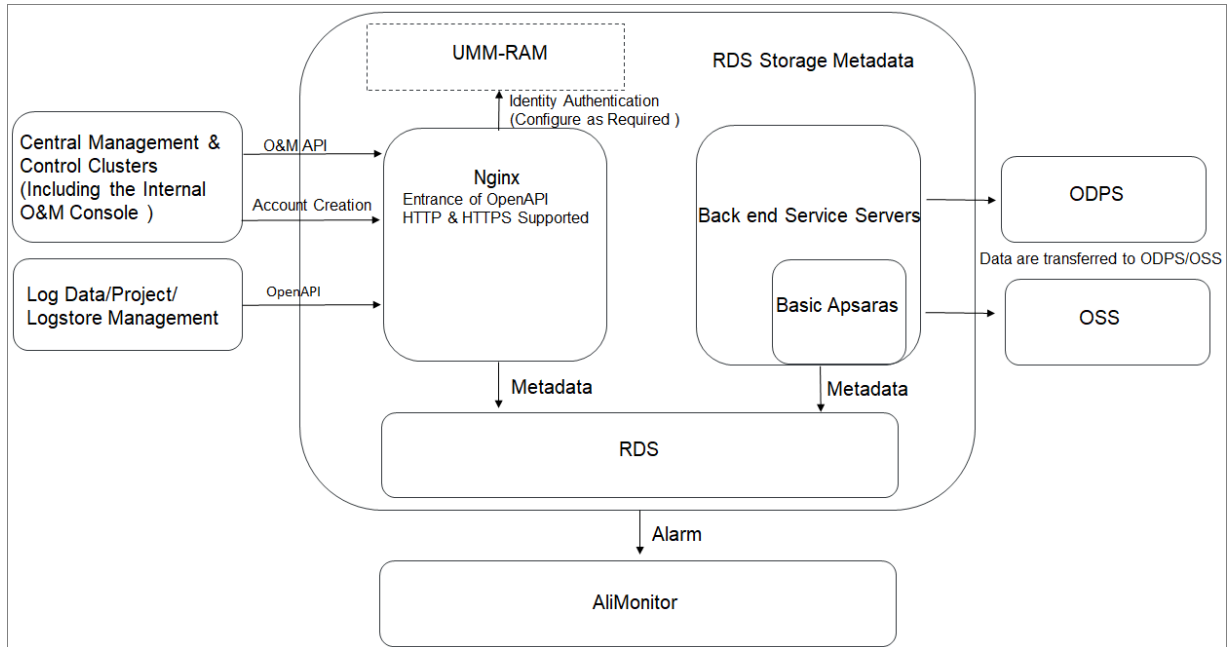
**The back-end service is a distributed process deployed across multiple machines. The service performs storage, indexing, and queries on Logstore data in real time. The overall characteristics of the back-end service are as follows:**

- **High data security**
  - **Each log is saved to three copies stored on different machines.**
  - **Data is automatically recovered in the cases of disk damage or machine downtime.**
- **Stable service**
  - **Logstore is automatically migrated in the cases of process crashes or machine downtime.**
  - **Automatic server load balancing ensures that traffic is distributed evenly among different machines.**
  - **Strict quotas prevent some unexpected or incorrect operations of a single user from affecting other users.**
- **Scale-out**
  - **A shard is the basic unit for scale-out.**
  - **You can add shards as needed to increase throughput.**

## 17.3.2 System architecture

The following figure shows the architecture of Log Service.

Figure 17-1: Architecture



- The console and open APIs are located on the left side. They are used to interact with external modules.
- The core modules in the middle include:
  - UMM and RAM: account management
  - RDS: metadata storage
  - NGINX: front-end servers
  - Log Service background: back-end business servers

# 18 Apsara Stack Security

---

## 18.1 What is Apsara Stack Security?

**Apsara Stack Security is a solution that provides Apsara Stack with a full suite of security features, such as network security, server security, application security, data security, and security management.**

**In today's cloud computing environment, new technologies are developed every day. Conventional network border protection methods based on detection technologies cannot ensure the security of cloud services. Apsara Stack Security combines the powerful data analytics capabilities of Alibaba Cloud with the professional expertise of the security operations team. It provides integrated security protection services at the network layer, application layer, and host layer.**

**Apsara Stack Security Standard Edition provides the following features:**

- **Threat Detection Service:** Captures network security data and analyzes the security situation. Performs correlation analysis on security events based on big data. Displays the detected security events and threats and provides security reports. Allows you to manage server assets, NAT assets, and vulnerabilities. Provides Web attack blocking and brute-force attack blocking functions.
- **Server Security:** Provides functions, such as threat prevention, intrusion detection, and log retrieval, to help the security administrator protect servers.
- **Application Security:** Provides Web Application Firewall (WAF) to help the security administrator protect the applications in Apsara Stack.
- **Network Security:** Helps the security administrator manage the internal and external network security of Apsara Stack.
- **System Management:** Provides account management, alert settings, and global settings.

## 18.2 Advantages

**Since the enforcement of China Internet Security Law, Regulations on Critical Information Infrastructure Security Protection and Cloud Security Classified Protection Standard 2.0 have been published. As a result, private cloud platforms**

**must pass the classified protection evaluation to ensure the security of cloud systems. Increasing security threats such as attacker intrusions and ransomware have led to the rising needs for security issue detection and prevention.**

**At the network perimeter of Apsara Stack, Apsara Stack Security uses a traffic security monitoring system to detect and block network-layer attacks in real time. It detects and removes Trojans and malicious files on servers to prevent attackers from exploiting the servers. In addition, Apsara Stack Security can block brute-force attacks and send alerts on unusual logons. This prevents attackers from stealing or destroying business data after logging on the system with weak passwords.**

#### **In-depth defense system**

**Apsara Stack Security comprises multiple functional modules. These modules work together to provide in-depth defense on the Apsara Stack network perimeter , within the Apsara Stack network, and on the Elastic Compute Service (ECS) instances in Apsara Stack. To help you manage security risks of Apsara Stack in a centralized manner and in real time, Apsara Stack Security provides a unified security management system. This system allows you to manage the security policies in all security protection modules and perform association analysis on the logs.**

**The security protection modules provided by Apsara Stack Security cover network security, server security, application security, and threat analysis. Based on a management center that can integrate the security information from all modules, Apsara Stack Security can accurately detect and block attacks. In this way, Apsara Stack Security protects your business systems in the cloud against intrusions.**

#### **Security solutions completely integrated with the cloud platform**

**Apsara Stack Security is a product born from ten years of protection experience . After a decade of experience in providing security operations services for the internal businesses of Alibaba Group and six years of safeguarding the Alibaba Cloud security operations, Alibaba has obtained considerable security research achievements, security data, and security operations methods, and has built a professional cloud security team. Apsara Stack Security brings together the rich experience of these experts to develop the sophisticated systems that provide enhanced security for cloud computing platforms. This product can protect the**

cloud platform, cloud network environments, and cloud business systems of Apsara Stack users.

The components of Apsara Stack Security are software-defined, with a full hardware compatibility. With these components, you can implement elastic cloud computing services based on quick deployment, expansion, and implementation . The protection modules on the cloud network perimeter or in the cloud network adopt the bypass architecture, which completely fits the cloud businesses and has the minimal adverse impacts on the cloud businesses. The protection modules running on the ECS instances are all virtualized to fit the flexibility of the ECS instances.

#### User security situation awareness

The cloud platform provides services for users. In Apsara Stack Security console, a user can view the security protection data, generate security reports, and enable SMS and email alerts by configuring external resources.

#### Security capability output

Apsara Stack Security has accumulated a large number of protection policies over the last several years. The service has protected millions of users from hundreds of thousands of attacks every day. This has generated a large amount of security protection data. Apsara Stack Security analyzes over 10 TB of this data every day . The analysis results are used to enhance the fundamental security capabilities , such as the malicious IP library, malicious activity library, malicious sample library, and vulnerability library. These capabilities are applied in the protection modules of Apsara Stack Security to enhance your business security.

## 18.3 Architecture

Apsara Stack Security consists of Apsara Stack Security Standard Edition and optional security services.

#### Apsara Stack Security Standard Edition

- **Traffic Security Monitoring:** This module is deployed on the network perimeter of Apsara Stack. It allows you to inspect and analyze each inbound or outbound packet of an Apsara Stack network by traffic mirroring. The analysis results are used by other Apsara Stack Security modules.



- **Server Intrusion Detection:** This module collects information and performs detection through the client deployed on physical servers. It detects file tampering, suspicious processes, suspicious network connections, suspicious port listening, and other suspicious activities on all servers in the Apsara Stack environment. This helps you detect server security risks in time.
- **Server Guard:** This module provides security protection features such as vulnerability management, baseline check, intrusion detection, and asset management for Elastic Compute Service (ECS) instances through log monitoring, file analysis, and signature scanning.
- **Web Application Firewall (WAF):** This module protects Web applications against common Web attacks defined by Open Web Application Security Project (OWASP), such as Structured Query Language (SQL) injections, cross-site scripting (XSS), exploit of Web server plugin vulnerabilities, Trojan upload, and unauthorized access. It blocks a large number of malicious visits to avoid website data leaks. This ensures the security and availability of your website.
- **Threat Detection Service (TDS):** This service collects traffic data and server information and detects potential intrusions or attacks through machine learning and data modeling. It detects vulnerability exploitation and new virus attacks launched by advanced attackers, and shows you the information of ongoing attacks, enabling business security visualization and awareness.

Apsara Stack Security Standard Edition also provides on-premises security operations services. On-premises security operations services help you make better use of the features of Apsara Stack products and Apsara Stack Security to ensure your application security.

Security operations services include pre-release security assessment, access control policy management, Apsara Stack Security product configuration, periodic security check, routine security inspection, and urgent event handling. These services cover the entire lifecycle of your businesses in Apsara Stack. On-premises security operations services help you create a security operations system for cloud businesses. This system enhances the security of application systems and ensures the security and stability of your businesses.

#### Optional security services

You can also choose the following optional service modules based on your own business needs to enhance your system security.

- **DDoS Traffic Scrubbing:** This module detects and filter out Distributed Denial of Service (DDoS) attack traffic to block DDoS attacks.

## 18.4 Features

### 18.4.1 Apsara Stack Security Standard Edition

#### 18.4.1.1 Threat Detection Service

Threat Detection Service (TDS) is a big data security analysis system developed by the Alibaba Cloud security team.

This module analyzes traffic data and server information to detect possible intrusions or attacks through machine learning and data modeling. It detects vulnerability exploits and new virus attacks launched by advanced attackers, and shows you the information about ongoing attacks, enabling you to monitor the security of your business systems.

#### Features

The following table describes the features that TDS provides.

Feature		Description
Security situation overview	Security situation overview	Provides overall security information, including the number of emergencies, attacks on the current day, flaws on the current day, attack trends, latest threat analysis, latest intelligence, and protected assets information.
	Screens	Provides the following screens for displaying security information: <ul style="list-style-type: none"><li>• Map-based traffic data screen.</li><li>• Server security screen.</li></ul>

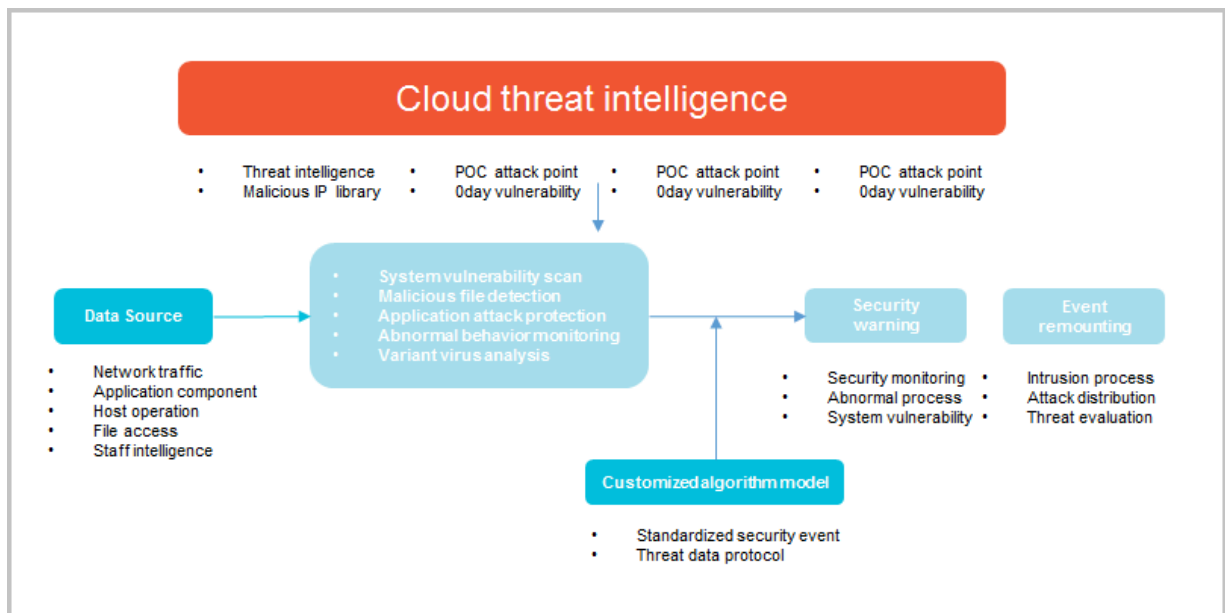
Feature		Description
Event analysis	Security event analysis	<p>Uses big data algorithms and models to detect the following security events in the cloud:</p> <ul style="list-style-type: none"> <li>• <b>Zombie activities:</b> A server becomes a zombie when it is controlled by an attacker and launches distributed denial of service (DDoS ) attacks on other servers.</li> <li>• <b>Brute-force attacks:</b> An attacker logs on to a server through brute-force attacks.</li> <li>• <b>Backdoors:</b> The WannaCry ransomware, unusual MySQL scripts, or webshells are detected.</li> <li>• <b>DDoS attacks:</b> A server encounters DDoS attacks.</li> <li>• <b>Hacking tools:</b> The logon credentials of a server are stolen, and hacking tools and attacks are detected.</li> <li>• <b>Suspicious network connections:</b> An attacker uses PowerShell to download suspicious files, runs suspicious VBScript commands, downloads malicious files or scripts to Linux servers, or runs commands in reverse shells.</li> <li>• <b>Unusual traffic:</b> A mining process is running.</li> </ul>
Traffic Security Monitoring	Traffic data collection and analysis	<ul style="list-style-type: none"> <li>• 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.</li> <li>• Collects the traffic data of a Classless Inter-Domain Routing (CIDR) block or IP address, including traffic on the current day, traffic in the last 30 days, traffic in the last 90 days, and the queries per second (QPS).</li> </ul>
	Malicious server identification	<p>Detects attacks launched by internal servers to identify controlled internal servers.</p>

Feature		Description
	Web application protection	Uses a bypass to block common attacks on Web applications at the network layer based on default Web attack detection rules. The attacks that can be blocked include Structured Query Language (SQL) injections, code and command execution, Trojan scripts, file inclusion attacks , and exploitation of upload vulnerabilities and common content management system (CMS) vulnerabilities.
	Unusual traffic detection	Uses a bypass in traffic mirroring mode to detect the unusual traffic that has exceeded the scrubbing threshold and reroutes the traffic to the DDoS Traffic Scrubbing module. The traffic rate (Unit: Mbit/s), packet rate (Unit: PPS ), HTTP request rate (Unit: QPS), or number of new connections can be set as the threshold.

How it works

The following figure shows how TDS works.

Figure 18-1: How TDS works



- **Big data security analysis platform**
  - **Network:** TDS uses the HTTP requests and responses collected by the Traffic Security Monitoring module to create HTTP logs, and uses big data models to analyze the logs and detect security events and threats.
  - **Server:** TDS uses the rules engine to analyze the server process data collected by Server Guard and detect security events and threats.
- **Security event display**
  - **Security events reported by Server Guard**
  - **Server security events discovered by server process analysis based on the rules engine**
  - **Network security events discovered by HTTP log analysis based on big data models**

#### Benefits

**TDS has the following benefits:**

- **Threat analysis based on big data**

TDS provides analysis and computing of petabyte-level big data and collects all security data and threat intelligence of the entire network. It also uses machine learning technologies to create complete and smart security threat models that can be used in the application scenarios of millions of users.

TDS focuses on the security trends and new threats that are faced by users of cloud computing in data centers, such as targeted Web application attacks, brute-force attacks, and system intrusions. It defends user systems against these threats from different fields.

- **Screens**

Based on Internet visualization technologies, TDS displays the results of big data threat analysis in graphs on screens to support security decision making on Apsara Stack.

#### 18.4.1.2 Traffic Security Monitoring

The Traffic Security Monitoring module is an Apsara Stack Security service that can detect attacks within milliseconds.

By performing in-depth analysis on the traffic packets mirrored from the Apsara Stack network ingress, this module can detect various attacks and unusual

activities in real time and coordinate with other protection modules to implement defenses. The Traffic Security Monitoring module provides a wealth of information and basic data support for the entire Apsara Stack Security defense system.

## Features

The following table describes the features that the Traffic Security Monitoring module provides.

Feature	Description
<b>Traffic data collection and analysis</b>	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.
<b>Unusual traffic detection</b>	Uses a bypass in traffic mirroring mode to detect the unusual traffic that has exceeded the scrubbing threshold and reroutes the traffic to the DDoS Traffic Scrubbing module. The traffic rate (Unit: Mbit/s), packet rate (Unit: PPS), HTTP request rate (Unit: QPS), or number of new connections can be set as the threshold.
<b>Malicious server identification</b>	Detects attacks launched by internal servers to identify controlled malicious servers.
<b>Web application protection</b>	Uses a bypass to block common attacks on Web applications at the network layer based on default Web attack detection rules. The attacks that can be blocked include Structured Query Language (SQL) injections, code and command execution, Trojan scripts, file inclusion attacks, and exploitation of upload vulnerabilities and common content management system (CMS ) vulnerabilities.
<b>Suspicious TCP connection blocking</b>	Uses a bypass to send TCP RST packets to the server and the client to block layer-4 TCP connections.
<b>Network log recording</b>	Records UDP and TCP traffic logs and the Request and Response logs of HTTP queries. Threat Detection Service (TDS) uses these logs for big data analysis.

## How it works

The Traffic Security Monitoring module collects data, processes the data, and then generates data processing results. It uses sockets to exchange data.

- **Collection:** The module collects traffic data through multiple high-performance PCs with dual-port 10GE network interface controllers (NICs).

- **Processing:** Traffic from an IP address may pass through multiple collectors. Traffic data must be consolidated to generate usable information.
- **Output:** The module stores and provides the consolidated traffic data.

### 18.4.1.3 Server Guard


Server Guard provides security protection measures such as vulnerability management, baseline check, intrusion detection, and asset management for Elastic Compute Service (ECS) instances by means of log monitoring, file analysis, and feature scanning.

Server Guard uses the client-server model. To protect the security of ECS instances in real time, Server Guard clients work with the Server Guard server to monitor attacks, vulnerabilities, and baseline configurations at the system layer and the application layer on the ECS instances.

#### Features

The following table describes the features provided by Server Guard.

Category	Feature	Description
Vulnerability management	Linux software vulnerability detection and fixes	Detects vulnerabilities recorded in the official database of Common Vulnerabilities and Exposures (CVE) in software such as SSH, OpenSSL, and MySQL based on the software versions, and provides vulnerability information and solutions.
	Windows vulnerability detection and fixes	Detects critical Windows vulnerabilities on your ECS instances based on the latest vulnerability information released by Microsoft, and provides Windows patches to fix the vulnerabilities, such as the Server Message Block (SMB) remote code execution vulnerability.


**Note:**  
 By default, only critical vulnerabilities are reported. You can manually check for security updates and detect low-risk vulnerabilities.

Category	Feature	Description
	<b>Web CMS vulnerability detection and fixes</b>	<p>Detects Web content management system (CMS) vulnerabilities based on the security intelligence provided by Alibaba Cloud by scanning directories and files. This feature also provides patches developed by Apsara Stack Security to fix vulnerabilities in software such as WordPress and Discuz!, and allows you to undo vulnerability fixes.</p>
	<b>Configuration and component vulnerability detection</b>	<p>Detects vulnerabilities that cannot be detected by software version comparison or file vulnerability scanning, and identifies critical configuration vulnerabilities in software, such as configuration and component vulnerabilities including unauthorized access to Redis and ImageMagick vulnerabilities.</p>
<b>Baseline check</b>	<b>Account security baseline check</b>	<ul style="list-style-type: none"> <li>• Detects SSH, RDP, FTP, MySQL, PostgreSQL, and SQL Server accounts with weak passwords.</li> <li>• Detects the at-risk accounts of your ECS instances, such as suspicious hidden accounts and cloned accounts.</li> <li>• Checks the compliance of the password policy of Linux servers.</li> <li>• Detects accounts without passwords on the ECS instances.</li> </ul>



Category	Feature	Description
	<b>System configuration check</b>	<p>Checks the system group policies, baseline logon policies, and registry configuration risks, including:</p> <ul style="list-style-type: none"> <li>• Suspicious auto-startup items in the scheduled tasks of Linux servers.</li> <li>• Auto-startup items on Windows servers.</li> <li>• Sharing configurations of the system.</li> <li>• SSH logon security policies of Linux servers.</li> <li>• Account-related security policies on Windows servers.</li> </ul>
	<b>Database security baseline check</b>	Checks whether the Redis service on an ECS instance is exposed to the public network, whether unauthorized access vulnerabilities exist, and whether suspicious data is written to important system files.
	<b>Benchmark compliance check</b>	Checks whether the system baseline complies with the latest Center for Internet Security (CIS) CentOS Linux 7 Benchmark.
<b>Intrusion detection - unusual logons</b>	<b>Disapproved logon location alerts</b>	Automatically records all logons, and determines the approved logon cities based on the usual logon locations . It also generates alerts on logons in disapproved locations. You can customize the approved logon cities.
	<b>Disapproved logon IP alerts</b>	Generates alerts on logons through IP addresses that are not whitelisted after a logon IP whitelist is created.
	<b>Disapproved logon time alerts</b>	Generates alerts on logons within disapproved time ranges after the approved logon time range is set.
	<b>Disapproved logon account alerts</b>	Generates alerts on logons with disapproved accounts after the approved logon account list is created.

Category	Feature	Description
	Brute-force attack detection and blocking	Detects and blocks brute-force attacks in real time. Both SSH and RDP brute-force attacks can be detected.
Intrusion detection - webshells	Webshell detection	Uses an Alibaba-developed webshell detection engine to detect and remove webshells on your ECS instances. Both scheduled and real-time scans are supported. It detects webshells written in Active Server Pages (ASP), Hypertext Preprocessor (PHP), and Java Server Pages (JSP), and allows you to manually quarantine these webshell files.
Intrusion detection - suspicious processes	Suspicious process activity detection	Detects suspicious process activities such as reverse shells, Java processes running CMD commands, and unusual file downloads using bash.
Asset management	Asset grouping	Allows you to group your ECS instances into a maximum of four layers, filter assets by region, online status, or other features, and manage the asset tags.
	Asset fingerprints	<ul style="list-style-type: none"> <li>• <b>Listening port:</b> collects and displays the listening port information and records changes. This allows you to easily check the listening port status.</li> <li>• <b>Account:</b> collects the account and permission information to discover privileged accounts and detect privilege escalations.</li> <li>• <b>Process:</b> collects and displays the information about the processes through snapshots to list normal processes and detect suspicious processes.</li> <li>• <b>Software:</b> lists the software installation information so that the affected assets can be quickly located when a critical vulnerability is exploited.</li> </ul>

Category	Feature	Description
Server logs	Log retrieval	<ul style="list-style-type: none"> <li>• <b>Previous logon logs:</b> records successful system logons.</li> <li>• <b>Brute-force attack logs:</b> records failed system logons.</li> <li>• <b>Process snapshot logs:</b> records the information about the running processes on the server at a specific time.</li> <li>• <b>Listening port snapshot logs:</b> records the listening port information on the server at a specific time.</li> <li>• <b>Account snapshot logs:</b> records the information about logon accounts on the server at a specific time.</li> <li>• <b>Process startup logs:</b> records the process startup information on the server.</li> <li>• <b>Network connection logs:</b> records the network connections started by the server.</li> </ul>

#### How it works

Server Guard uses the client-server model. The client is installed on ECS instances. The client communicates with the server through a TCP persistent connection and uses HTTP to obtain scripts, rules, and installer packages from the server.

The client can be used in Windows or Linux. It can automatically connect to the server for online updates.

The key features of Server Guard work as follows:

- **Vulnerability management:** The client collects the ECS instance information, including component information, software versions, file information, and registry information. Then, the client checks whether the information matches the vulnerability detection rules provided by the server. The information that matches the rules will be sent to the server for further analysis. The detected vulnerabilities will be displayed in the Server Guard console. You can fix vulnerabilities in the console or by calling API operations. After receiving the vulnerability patches from the server, the client on the vulnerable ECS instance

automatically fixes the vulnerabilities and synchronizes the vulnerability status to the server.

- **Baseline check:** When you manually start a check or a periodic check is triggered, the Server Guard server sends a baseline check request to the client. The client then collects the server information according to the check policy and compares the information with the security baseline. Check items that do not comply with the baseline are labeled as at-risk items and reported to the server.
- **Unusual logon detection:** The client monitors the logon logs of the server system in real time. In a Linux system, the `/var/log/secure` and `/var/log/auth.log` files are also monitored. All failed and successful logons are recorded. Unusual logons or brute-force attacks will be reported to the server.
- **Webshell detection:** The client uses an Alibaba-developed dynamic webshell detection engine to detect complex webshells. It then restores these webshells to an identifiable status to analyze the hidden webshell activities. This prevents webshells from bypassing the detection due to the use of static detection rules.
- **Suspicious process detection:** The Server Guard server uses a data analysis rules engine to analyze the server process data collected by the client. By doing so, the server can detect suspicious processes such as reverse shells, mining processes, DDoS Trojans, worms, viruses, and hacking tools.
- **Log collection:** The client collects logs such as processes logs and network logs.

## Scenarios

Server Guard is applicable to server security protection in the following scenarios:

- **Use common software for website building**

In this scenario, attackers may intrude servers by exploiting vulnerabilities in common software. You can use Server Guard to detect and fix vulnerabilities.

- **Use Web application services**

Attackers may steal website data through both internal and external Web services. You can use Server Guard to prevent attackers from launching attacks or controlling your servers.

### 18.4.1.4 Server Intrusion Detection

The Server Intrusion Detection module collects information through the client program installed on a physical server. It detects file tampering, suspicious

processes, suspicious network connections, suspicious port listening, and other activities on all servers in the Apsara Stack environment. This helps you detect server security risks in time.

#### Features

The Server Intrusion Detection module provides the following features:

Feature	Description
Key directory integrity check	Detects file tampering in the <code>/etc/init.d</code> path in the server system and generates alerts.
Suspicious process alert	Detects suspicious processes such as XOR DDoS trojans, Bill Gates DDoS trojans, and MinerD mining processes, and generates alerts.
Suspicious port listening alert	Detects new port listening tasks in time, and generates alerts.
Suspicious network connection alert	Detects connections to the public network actively initiated by internal network servers, and generates alerts.

#### How it works

The client program is installed on a physical server and collects information in real time, including the key directories, processes, open ports, and network connections. Then it detects suspicious events using rule and signature matching, and reports the suspicious events to the user.

### 18.4.1.5 Web Application Firewall

Web Application Firewall (WAF) protects the Web applications of cloud users against common Web attacks.

Different from traditional web application firewalls, Apsara Stack WAF uses intelligent semantic analysis algorithms to identify Web attacks. WAF also integrates a learning model to enhance its analysis capability so that it can meet your daily security protection requirements without relying on traditional rule libraries.

WAF protects the traffic of businesses on HTTP and HTTPS websites. In the WAF console, you can import certificates and private keys to enable end-to-end encryption. This prevents the interception of business data on the links.

WAF not only prevents common Web application attacks defined by Open Web Application Security Project (OWASP) but also mitigates HTTP flood attacks. In addition, WAF allows you to customize protection policies based on the businesses of your website to block malicious Web requests.

## Features

The following table describes the features provided by WAF.

Feature	Description
Protection against common Web attacks	<p>Detects Structured Query Language (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.</p> <p>WAF 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, enable or disable each attack detection module separately, and set the detection granularity.</p>
HTTP flood mitigation	<p>Allows you to set access frequency control rules for domain names and URLs to restrict the access frequency of IP addresses or sessions that meet the criteria or block these IP addresses or sessions.</p> <p>Restricts the access frequency of known IP addresses or sessions or block these IP addresses or sessions.</p> <p>The HTTP flood mitigation rules do not apply to IP addresses or sessions that have been added to the whitelist.</p>
Custom and precise access control	<p>Supports precise access control based on the following HTTP contents or their combinations: URI, GET parameters, decoded path, HOST header, complete cookie, POST parameters, complete body, HTTP status code, and response content.</p>

## How it works

WAF performs protocol parsing and in-depth decoding on the Web access traffic . It then calls the access control, rule detection, and semantic analysis engines to analyze the traffic and determines whether to allow or block the traffic based on the preset policies. Besides, WAF provides a good human-machine interaction interface for administrators to adjust protected websites and security policies.

## Scenarios

WAF can be used for Web application protection in fields such as government, finance, insurance, e-commerce, online to offline (O2O), Internet Plus, and games. It provides the following features:

- Prevents website data leaks caused by SQL injections.
- Mitigates HTTP flood attacks by blocking a large number of malicious requests. This ensures the availability of your website.
- Prevents website defacement arising from Trojans to ensure the credibility of your website.
- Provides virtual patches that enable quick fix for newly discovered vulnerabilities.

### 18.4.1.6 On-premises security operations services

To ensure the stability, reliability, security, and regulatory compliance of the cloud platform, Apsara Stack Security Standard Edition provides multiple security products and on-premises security operations services to ensure the availability, confidentiality, and integrity of the systems and data of users. Security operations services are indispensable in the security system. The combination of security products and security operations services gives full play to the security features of both Apsara Stack products and Apsara Stack Security products, and enhances the security of the Apsara Stack network environment from both technology and management aspects.

On-premises security operations services aim to help users use the security features of both Apsara Stack products and Apsara Stack Security products to protect the user applications. Security operations services include services that cover the entire security lifecycle of Apsara Stack user businesses, such as pre-release security assessment, access control policy optimization, periodic security assessment, routine security inspection, and emergency response. These services

help users create a cloud security operations system to enhance the application system security and ensure secure and stable businesses.

## Services

On-premises security operations services are as follows:

Table 18-1: On-premises security operations services

Category	Service	Description
User business security operations	User asset research	With the authorization of a user, this service periodically researches the cloud businesses of the user and develops a business list containing information such as the business system name, ECS, RDS, IP address, domain name, and owner.
	New business security assessment	<ul style="list-style-type: none"> <li>Before a user migrates a new business system to the cloud, this service detects system vulnerabilities and application vulnerabilities in the new business system using both automation tools and manual operations.</li> <li>Provides advice and verification on vulnerability fixes.</li> </ul>
	Periodic business security assessment	<ul style="list-style-type: none"> <li>Periodically uses automation tools to detect system vulnerabilities, application vulnerabilities, and security risks in running businesses.</li> <li>Provides advice on handling detected risks, including but not limited to security policy settings, patch updates, and application vulnerability handling.</li> </ul>
	Access control management	Provides inspection and guidance on applying access control policies when a new business is migrated to the cloud.
	Access control routine inspection	Periodically checks for access control risks of user businesses.



Category	Service	Description
	Security risk routine inspection	Monitors and inspects security events in Apsara Stack Security. Informs the user of verified events and provides advice on event handling.
Apsara Stack Security operations	Rule update	Periodically updates the rule libraries of Apsara Stack Security products.
	Product integration	<ul style="list-style-type: none"> <li>Provides support for integrating Apsara Stack Security products with the application systems of users.</li> <li>Helps users customize and optimize security policies.</li> </ul>
Security event response	Event alerts	Synchronizes recent security events information from Alibaba Cloud, and helps users remove the risks.
	Event handling	Handles urgent events such as attacker intrusions.

#### Service output

On-premises security operations services output the following documents:

- Weekly, monthly, and yearly service reports
- Asset lists
- System security check reports

#### SLA

The SLA terms of on-premises security operations services are as follows:

- Asset management: Update the asset list once a month.
- Security event response: Respond within 30 minutes during work hours.
- Security check:
  - Complete a pre-release security check within two workdays.
  - Perform a periodic security check once a quarter.

#### Duties

Partners authorized by Alibaba Cloud provide on-premises security operations services, and Alibaba Cloud provides service quality management and technical support.

Owner	Duties
Alibaba Cloud	<ul style="list-style-type: none"> <li>• Assign and manage tasks of service providers and on-premises engineers.</li> <li>• Assess the services provided by service providers and on-premises engineers.</li> <li>• Train service providers and on-premises engineers and provide technical support.</li> <li>• Provide project coordination and process and quality management.</li> </ul>
Service provider	<ul style="list-style-type: none"> <li>• Perform security check and routine inspection on the system of the user.</li> <li>• Provide advice on fixing vulnerabilities.</li> <li>• Maintain the access control policies of the user resources.</li> <li>• Update and maintain the security rules and policies of Apsara Stack Security.</li> <li>• Respond to security events.</li> <li>• Provide security technical support for users.</li> </ul>
User	<ul style="list-style-type: none"> <li>• Authorize service providers to perform security operations.</li> <li>• Follow the security advice to carry out the security plans on businesses.</li> <li>• Improve the security system.</li> </ul>

#### Risk control

The following measures are taken to control risks in on-premises security operations services:

Category	Risk Item	Measure
Engineer and organization qualification	Organization	Only Alibaba Cloud and authorized enterprises can provide security services.
	Engineers	All engineers must be assessed and trained by the Alibaba Cloud security team.
Confidentiality	Confidentiality agreements	All enterprise and individual service providers must sign a confidentiality agreement.
Service tool security	Tool selection	Only security tools specified by Alibaba Cloud are allowed.
	Tool use	Apply standard configurations to avoid risks in using the tools.

Category	Risk Item	Measure
Operation security	Operation procedure	Perform at-risk operations, such as scanning, in batches.
	Risk notification	Inform the users of risks in the operations, and provide risk avoidance and control methods. Perform operations only with the consent of the users.

## 18.4.2 Optional security services

In addition to the security services provided by Apsara Stack Security Standard Edition, multiple optional security services are also provided to meet various security needs. We recommend that you choose optional security services based on your business needs.

### 18.4.2.1 DDoS Traffic Scrubbing

Backed by its large-scale and distributed operating system and more than a decade of experience in defending against security attacks, Alibaba Cloud has designed and developed the DDoS Traffic Scrubbing module based on the cloud computing architecture to protect the Apsara Stack platform against large amounts of distributed denial of service (DDoS) attacks.

#### Features

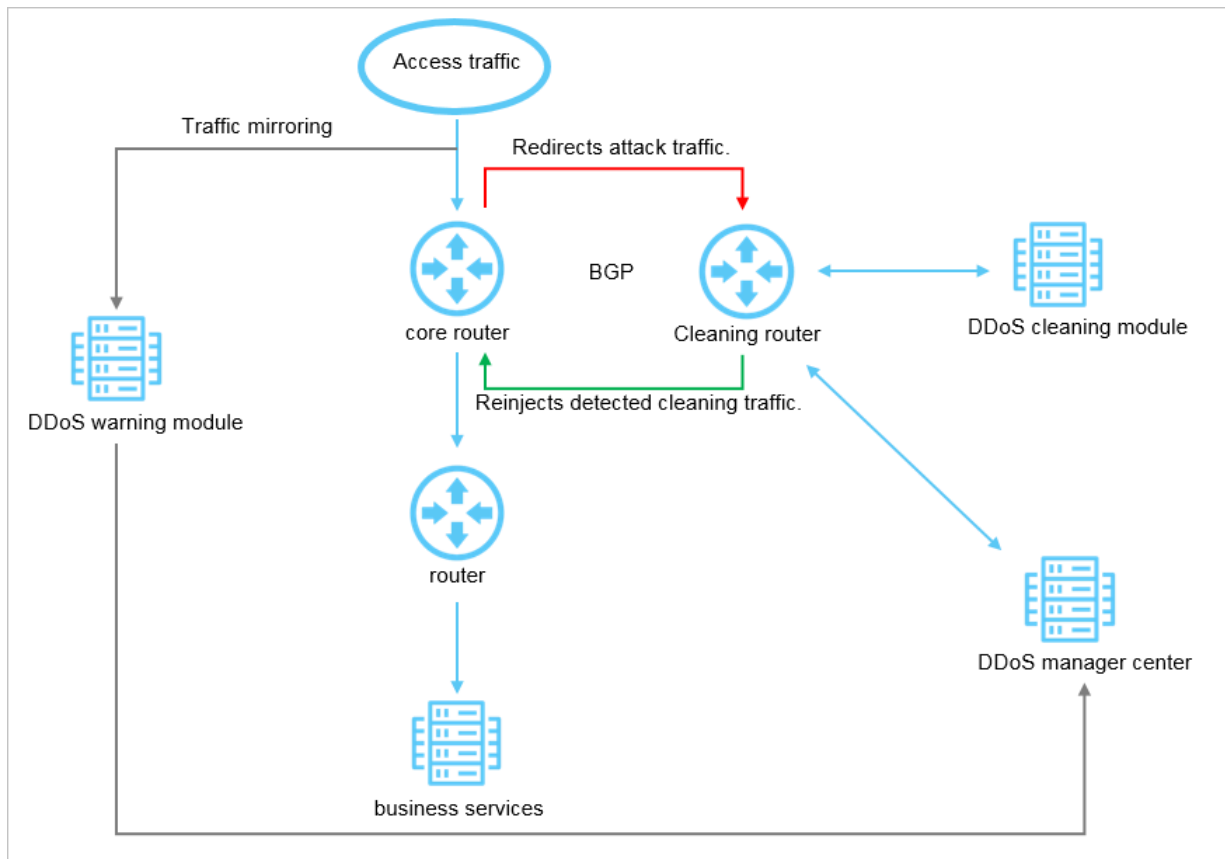
The following table describes the features provided by the DDoS Traffic Scrubbing module.

Feature	Description
Traffic scrubbing against DDoS attacks	Detects and prevents attacks such as SYN flood, ACK flood, ICMP flood, UDP flood, NTP flood, DNS flood, and HTTP flood.
DDoS attack display	Allows you to view DDoS attacks in the console and search for DDoS attacks by IP address, status, and event information.
DDoS traffic analysis	Allows you to monitor and analyze the traffic of a DDoS attack, and view the attack traffic protocol and the top 10 IP addresses that have launched most attacks.

## How it works

After the Traffic Security Monitoring module detects unusual traffic, the DDoS Traffic Scrubbing module reroutes, scrubs, and reinjects the traffic, as shown in [Figure 18-2: Traffic scrubbing](#). This mitigates DDoS attacks and ensures normal running of businesses.

Figure 18-2: Traffic scrubbing



The Traffic Security Monitoring module sends information about the detected DDoS attacks to the DDoS Traffic Scrubbing module. The DDoS Traffic Scrubbing module is connected to the border gateway device. When a DDoS attack is detected, this module configures a Border Gateway Protocol (BGP) path for the border gateway to reroute the attack traffic to the DDoS Traffic Scrubbing module. The DDoS Traffic Scrubbing module then scrubs the traffic based on the configured scrubbing policies, filters out unusual traffic, and reinjects the normal traffic to the border gateway.

**Note:**

Apsara Stack Security cannot scrub the traffic between internal networks.

## Advantages

**The DDoS Traffic Scrubbing module has the following feature advantages:**

- **Detection of all common DDoS attacks**

This module protects you from various DDoS attacks, such as HTTP flood, SYN flood, UDP flood, UDP DNS query flood, stream flood, ICMP flood, and HTTP GET flood, at the network layer, transport layer, and application layer. This module also informs you of the website defense status through real-time SMS messages.

- **Automatic response to attacks within one second**

This module uses the world leading attack detection and prevention technologies. It can complete the protection process within one second, covering attack discovery, traffic rerouting, and traffic scrubbing. This module triggers traffic scrubbing when the traffic scrubbing thresholds are violated or when DDoS attacks are detected during network behavior analysis. This reduces network jitter and ensures the availability of your businesses in the case of DDoS attacks.

- **High scalability and high redundancy of anti-DDoS capabilities**

With high scalability and high redundancy of the cloud computing architecture, this module can be easily scaled up to realize high scalability of anti-DDoS capabilities.

- **Bidirectional protection to avoid the abuse of cloud resources**

This module not only protects your system against external DDoS attacks but also detects resource abuse in your cloud environment. If any of your cloud resources in Apsara Stack is used to launch DDoS attacks, the Traffic Security Monitoring module will cooperate with Server Guard to restrict the network access of the hijacked resource and generate an alert.

### 18.4.2.2 Sensitive Data Discovery and Protection

**Sensitive Data Discovery and Protection (SDDP)** is a data security service used to detect and protect sensitive data in Apsara Stack big data services.

SDDP uses Alibaba's analysis capabilities in big data and related AI technologies to detect sensitive data precisely, and allows you to classify sensitive data based on your business requirements. Based on precise detection, SDDP can also dynamically and statically de-identify sensitive data, monitor data flows globally, and detect anomalous activities. SDDP provides visible, controllable, and standards-compliant

security protection for your sensitive data based on precise detection, precise detection, precise analysis, and effective protection. SDDP can protect sensitive data in a variety of Apsara Stack big data services, including MaxCompute, Object Storage Service (OSS), and Table Store.

## Features

The following table describes the features that SDDP provides.

Feature		Description
Sensitive data classification and detection	Data detection	A department administrator can authorize SDDP to scan and protect data assets of the department based on the business requirements . SDDP scans and monitors only the authorized data assets.
	Sensitive data classification	SDDP allows you to classify sensitive data in big data services such as MaxCompute, OSS, and Table Store. You can define sensitive data rules by using keywords or regular expressions.
	Sensitive data detection	SDDP provides built-in algorithms for discovering sensitive data, and can use file clustering, deep neural network, and machine learning to detect sensitive images, text, and fields.
Sensitive data permission management	Asset permission detection	SDDP allows you to view the authorization information about data assets and the accounts that have permissions to access data assets. The data assets include MaxCompute projects , MaxCompute tables, MaxCompute columns, MaxCompute packages, OSS bucket, Table Store instances, and Table Store tables.
	Account permission detection	SDDP allows you to view all accounts in a department and search for departments or accounts in fuzzy search mode. SDDP displays the relationships between departments and accounts in a hierarchical, visualized manner.
	Anomalous permission access detection	SDDP automatically detects anomalous permission access in big data services such as MaxCompute, OSS, and Table Store.

Feature		Description
Data flow and operation monitoring	Data flow monitoring	SDDP can monitor data flows among entities , such as data storage services (including MaxCompute, OSS, and Table Store), data transmission services (including Datahub and CDP), the data stream processing service Blink , external databases, and external files. SDDP provides visual diagrams to dynamically display data flows and anomalous activities. In addition , you can click a location where an anomalous activity has occurred in a diagram to go to the page for processing the anomalous activity.
	Anomalous data operation detection	SDDP can detect anomalous operations in big data services such as MaxCompute, OSS, and Table Store.
	Anomalous data flow detection	SDDP can detect anomalous data flows, including anomalous downloads, in big data services such as MaxCompute, OSS, and Table Store.
	Detection rule customization	SDDP allows you to customize rules for detecting anomalous data flows and anomalous data operations.
Anomalous activity processing	Anomalous activity configuration	SDDP allows you to configure the thresholds and rules for detecting anomalous activities, including anomalous data flows, anomalous permission access, and anomalous data operations.
	Anomalous activity processing	SDDP provides a built-in console for processing anomalous activities. You can search for anomalous activities by department, event type, account, processing status, and time of occurrence.
	Anomalous activity statistics	SDDP collects statistics on the processing status of anomalous activities, including anomalous data flows, anomalous permission access, and anomalous data operations, and then dynamically displays the statistics.

## Scenarios

- **Comply with applicable laws and regulations on personal information protection**

SDDP can detect personal information in a large amount of data, automatically mark risk levels of the personal information, and effectively detect data leaks. By using SDDP, enterprises can make sure that their systems comply with applicable laws and regulations on personal information protection.

- **Classify and protect enterprise sensitive data**

Based on specified rules, SDDP can detect sensitive data, manage data permissions, and detect anomalous activities, including anomalous data flows, anomalous permission access, and anomalous data operations. This allows enterprises to classify and protect their sensitive data.

- **Handle data leaks**

SDDP detects anomalous activities based on the specified rules and allows you to handle these events in a centralized manner. This allows enterprises to handle data leaks online, which provides effective support for security O&M.

## Benefits

As a data security module of Alibaba Cloud Security, SDDP can detect and protect sensitive data in both real-time computing services (including Blink, Datahub, and Table Store) and offline computing services (including MaxCompute and OSS). SDDP detects structured, semi-structured, and unstructured sensitive data based on the same standards. SDDP has the following benefits:

- **Precise detection**

Based on Alibaba's technologies accumulated through years in AI and expert systems, SDDP uses a built-in rule engine, natural language processing model, neural network model to precisely detect sensitive personal information, sensitive system configurations, and confidential documents in a large amount of data.

- **Closed-loop management**

SDDP forms a closed loop covering detection, protection, and handling to help enterprises effectively avoid risks.



- **Intelligent detection**

**Based on Alibaba's big data analysis and detection capabilities accumulated over the years, SDDP implements an intelligent, multi-level filtering model to detect anomalous activities effectively.**

- **Flexible definition**

**SDDP allows you to customize a variety of data based on your business requirements, including sensitive data detection rules, sensitive data definitions, and thresholds and rules for detecting anomalous activities.**

## 19 Key Management Service (KMS)

---

### 19.1 What is KMS?

**Key Management Service (KMS) is a secure and easy-to-use management service provided by Alibaba Cloud. KMS ensures the privacy, integrity, and availability of your keys at minimal costs.**

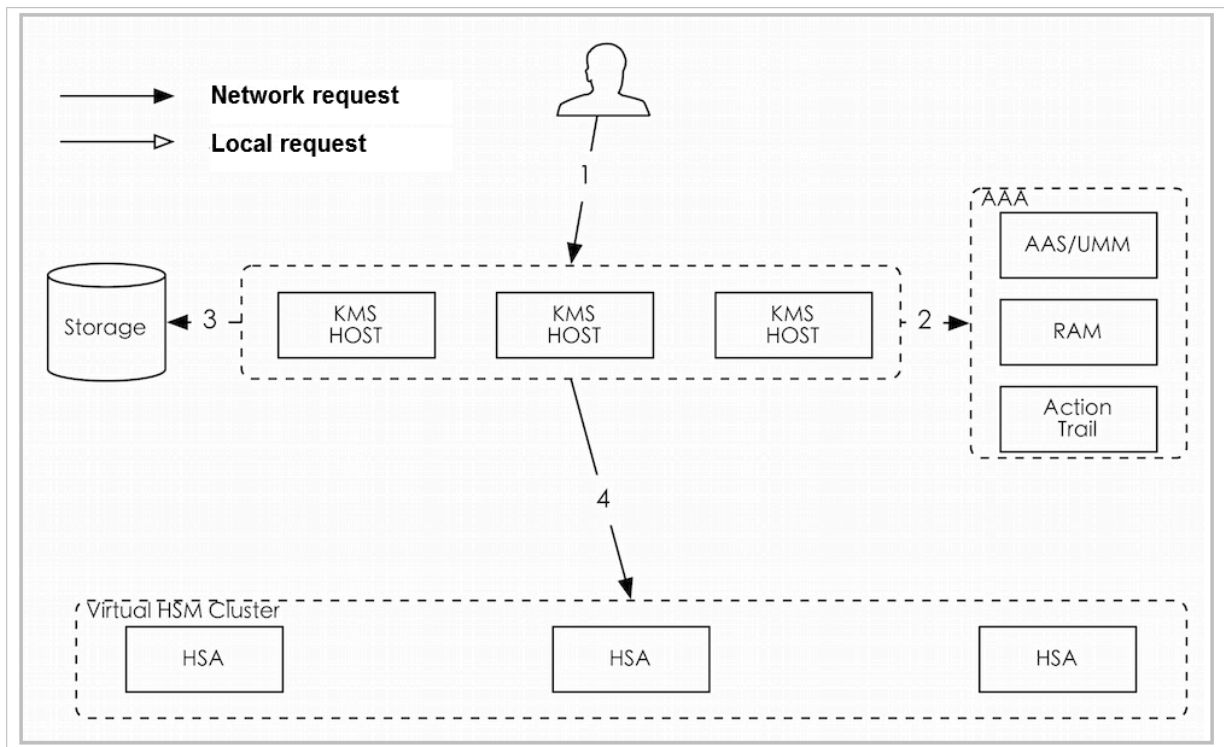
**KMS allows you to use keys securely and conveniently. You can focus on developing encryption and decryption solutions that best suit your needs.**

### 19.2 Architecture

**KMS is deployed across different regions. Each region provides the same functions , but the data of each region is mutually independent. Within a single region, KMS adopts a distributed architecture composed of multiple equivalent nodes. All nodes within a single region provide the same availability, allowing you to scale services based on your actual access needs.**

*Figure 19-1: Architecture* shows the architecture of KMS.

Figure 19-1: Architecture



**KMS consists of the following four modules:**

- **Storage**

**This module stores exported key tokens (EKTs) and other metadata.**

- **AAA**

**This module authenticates AccessKeys, authorizes RAM users, and audits disk invocation information.**

- **KMSHOST**

**This module processes user requests to call APIs.**

- **Hardware security appliance (HSA)**

**This module simulates the hardware security module (HSM) to process the cryptographic logic of KMS powered by Raft distributed storage and trusted computing technologies.**

## 19.3 Features

### 19.3.1 Convenient key management

You can use the APIs provided by KMS or the KMS console to facilitate managing your keys.

- You can disable and enable customer master keys (CMKs) at any time. After a CMK is disabled, the data encrypted by using this CMK cannot be decrypted.
- A schedule key deletion policy is used to delete CMKs. You can cancel scheduled key deletion at any time to reduce the potential impact of accidental operations.
- You can use Resource Access Management (RAM) to manage CMK permissions and separate the permissions for encryption and decryption operations.
- You can use Encryption Context to enhance control over keys and ciphertexts.

### 19.3.2 Envelope encryption technology

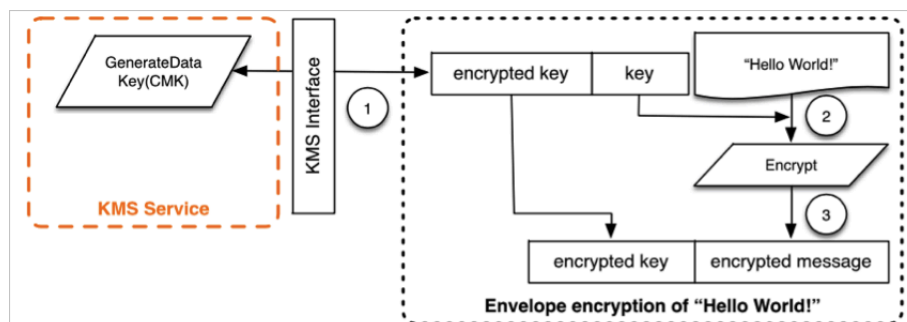
Although KMS provides the Encrypt API, KMS does not actually encrypt data. Instead, KMS manages CMKs and uses CMKs to encrypt and decrypt DEKs. You have to use DEKs to encrypt data yourself.

You can use your own DEK to encrypt data and then use the Encrypt API to encrypt your DEK. You can also use the GenerateDataKey API to obtain a DEK.

Encryption process

*Figure 19-2: Encryption flowchart* shows the envelope encryption process.

Figure 19-2: Encryption flowchart



The encryption process is as follows:

1. Use the specified CMK to generate a DEK and enveloped data key (EDK).

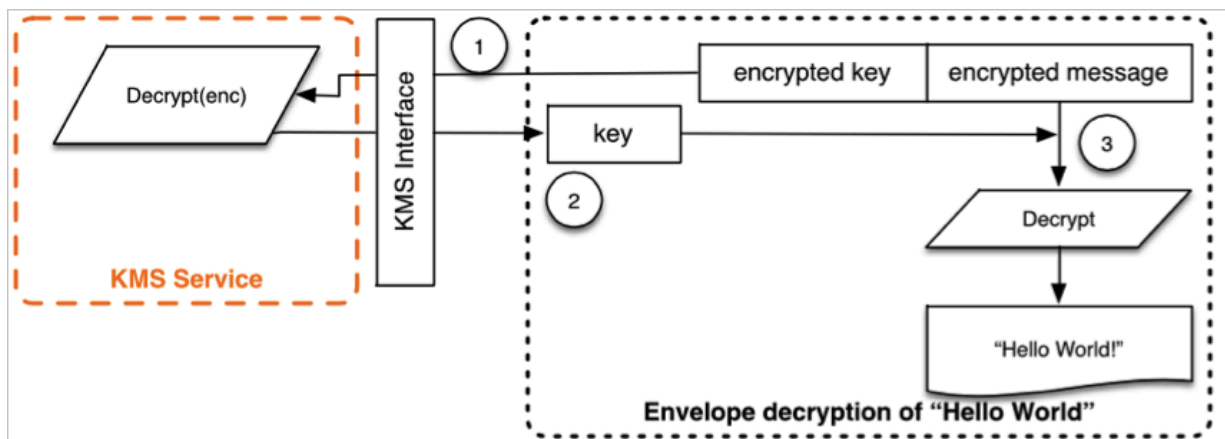
You can also use your own DEK and use the Encrypt API to obtain the corresponding EDK.

2. Use the DEK to encrypt the data and obtain the ciphertext.
3. Store the ciphertext together with the EDK in your device.

Decryption process

*Figure 19-3: Decryption flowchart* shows the envelope encryption process.

Figure 19-3: Decryption flowchart



The decryption process is as follows:

1. Use KMS to decrypt the EDK.
2. Obtain the DEK.
3. Use the DEK to decrypt the ciphertext and obtain the plaintext.

### 19.3.3 Secure key storage

KMS uses the following methods to ensure keys are stored securely:

- The CMK plaintext is stored only in the memory of the HSA module, while the CMK ciphertext is stored only in the KMS storage module.
- CMKs are encrypted by using DomainKeys managed by the HSA module. DomainKeys are updated on a daily basis.
- DomainKeys are encrypted for storage by using trusted computing technologies (such as TPMs) and stored based on a distributed storage protocol to ensure the high reliability of DomainKeys.

## 20 Apsara Stack DNS

---

### 20.1 What is Apsara Stack DNS?

**Apsara Stack DNS is a service that runs on Apsara Stack and resolves domain names**

- . Apsara Stack DNS resolves requested domain names based on preconfigured rules**
- . It helps redirect requests from clients to cloud services, business systems on enterprise intranets, and Internet services.**

**Apsara Stack DNS resolves and forwards DNS requests in VPCs.**

- Provides weight-based scheduling to meet your business needs for active zone -redundancy, active geo-redundancy, zone-disaster recovery, and geo-disaster recovery.**
- Provides domain name isolation by tenant to meet your business needs for department resource isolation.**

**You can perform the following operations on your VPC by using Apsara Stack DNS:**

- Access other ECS instances deployed in your VPC.**
- Access ECS instances provided by Apsara Stack.**
- Access custom enterprise business systems.**
- Access Internet businesses and services.**
- Establish network connections between Apsara Stack DNS and a user-created DNS by using a leased line.**

### 20.2 Benefits

**As a key network service, Apsara Stack DNS controls data flows that go through Apsara Stack, resolves domain names, balances server loads, and connects Apsara Stack to on-premises data centers. Apsara Stack DNS provides you with multiple solutions for cloud environment deployment, zone high availability, server load balancing, and disaster recovery to support your IT operations.**

**Enterprise domain name management**

**Apsara Stack DNS provides management and resolution services for enterprise domain names.**

- **Apsara Stack DNS supports DNS resolution and reverse DNS resolution for domain names of cloud service instances such as ECS instances.**
- **It also supports DNS resolution and reverse DNS resolution for your internal domain names.**
- **You can add, modify, and delete the following types of DNS records: A, AAAA, CNAME, MX, PTR, TXT, SRV, NAPTR, CAA, and NS.**
- **You can add multiple DNS records, such as A, AAAA, and PTR records, for one host. By default, resolution results include all matching records. The records can be randomly rotated for load balancing.**
- **You can add multiple DNS records, such as A, AAAA, and CNAME records, for one host. Resolution results are returned based on the weight of each record to implement global scheduling.**

#### Flexible networking

**Apsara Stack DNS can forward queries for enterprise domain names, allowing you to flexibly create or combine networks.**

- **Supports forwarding DNS queries for all domain names.**
- **Supports forwarding DNS queries for some domain names.**

#### Internet access from enterprise servers

**When the public network is accessible, Apsara Stack DNS supports recursive queries for public domain names and Internet domain names. With this feature, you can access Internet services using enterprise servers.**

#### Tenant isolation

**With VPC-based private zone management and resolution features, you can isolate DNS data and resolution by tenant.**

**You only need to purchase a set of DNS services to meet your needs for VPC isolation. You do not need to build multiple DNS systems, saving your capital expenditure.**

#### Unified management platform

**The DNS console is integrated into the Apsara Stack console. You can use the same account to manage both DNS and other Apsara Stack services. Apsara Stack DNS has the following benefits:**

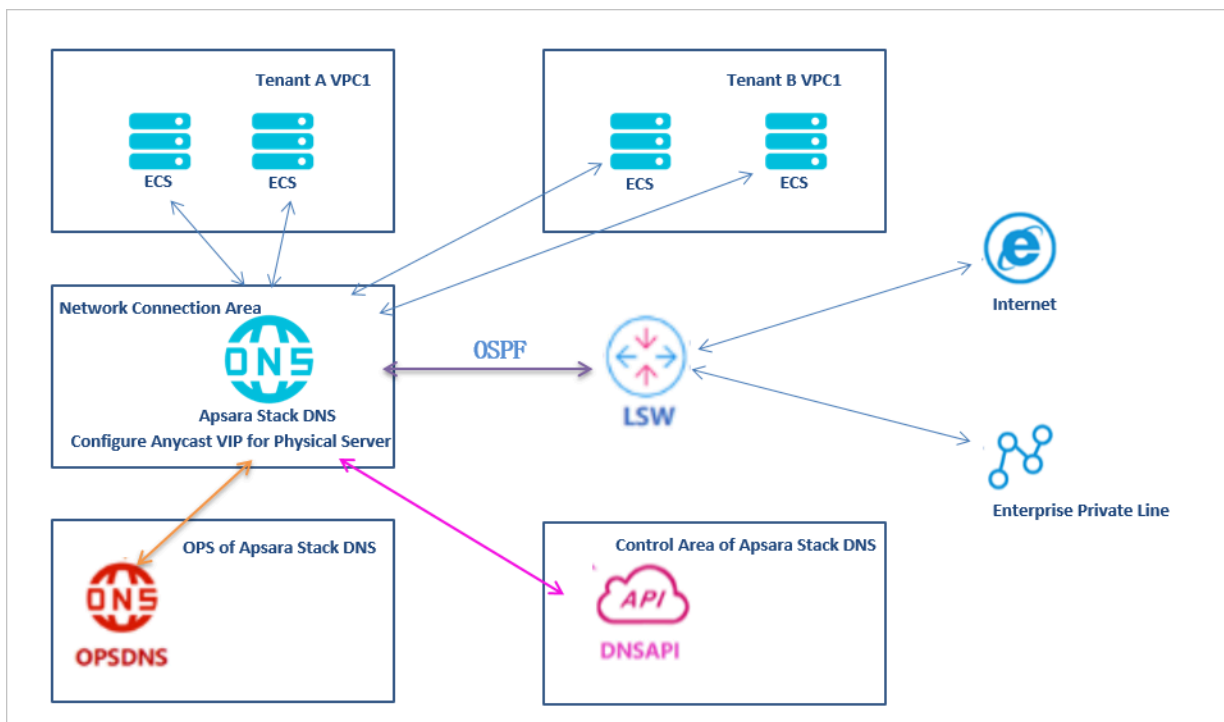
- Apsara Stack DNS supports web services for data and service management, which is easy to use.
- Apsara Stack DNS is deployed on clusters. You can add more clusters based on your needs.
- You can deploy Apsara Stack DNS in multiple zones. Apsara Stack DNS supports active zone-redundancy and zone-disaster recovery.
- Apsara Stack DNS is deployed in anycast mode. High availability and disaster recovery are enabled by default.

#### API operations

Apsara Stack DNS provides API operations and supports integration with other systems.

## 20.3 Architecture

Figure 20-1: Apsara Stack DNS architecture



#### Apsara Stack DNS architecture

- Deploys two or more physical servers for network connections.
- Uses two control interfaces for link aggregation. These interfaces are connected to an access switch (ASW). The gateway is the default gateway of the internal network.



- **Two service interfaces are connected to a LAN switch (LSW) over Equal-Cost Multi-Path (ECMP). These interfaces support Open Shortest Path First (OSPF) to advertise anycast VIP routes, and are connected to the Internet.**
- **The control system is deployed in a container in the control area.**

## 20.4 Features

### Internal domain name management

**Apsara Stack DNS provides management for internal domain names. You can register, search, and delete internal domain names and add descriptions. You can add, delete, and modify the following types of DNS records: A, AAAA, CNAME, MX, PTR, TXT, SRV, NAPTR, CAA, and NS. With the internal domain name resolution feature of Apsara Stack DNS, you can resolve domain names for servers in a VPC. The DNS endpoint is deployed in anycast mode. For disaster recovery, DNS can switch services between servers that are located in different data centers.**

### Domain name forwarding management

**Apsara Stack DNS can forward DNS queries for some or all domain names to other DNS servers.**

**Two forwarding modes are available: forward all requests with recursion and forward all requests without recursion.**

- **Forward all requests without recursion: Forwards DNS requests to the target DNS server. If the target DNS server cannot resolve the domain names or the request is timed out, a message is returned to the DNS client indicating that the query failed.**
- **Forward all requests with recursion: Forwards DNS requests to the target DNS server. If the target DNS server cannot resolve the domain names, the local DNS server is used to resolve them.**

### Recursive resolution

**With recursive resolution, you can resolve Internet domain names to access Internet services.**

### Tenant isolation (standard edition only)

**With VPC-based private zone management and resolution features, enterprises can isolate DNS data and resolution by tenant.**

## 21 API Gateway

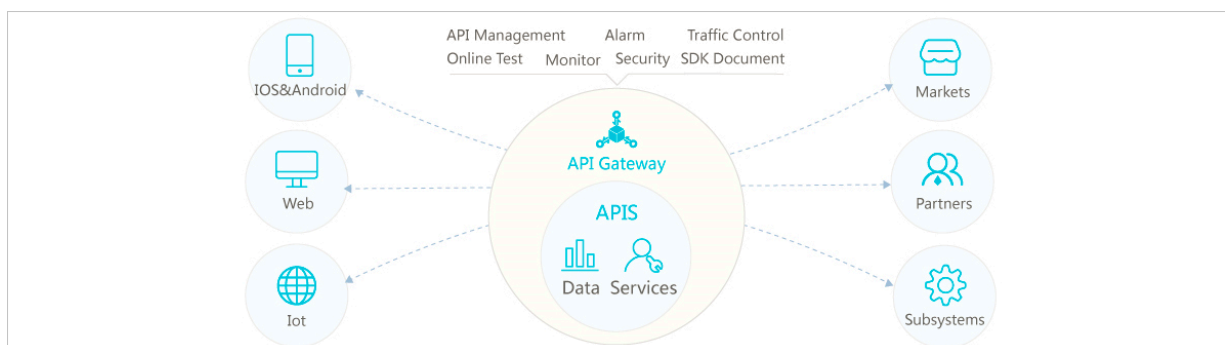
### 21.1 What is API Gateway?

API Gateway provides a comprehensive suite of API hosting services that help you share capabilities, services, and data with partners in the form of APIs. API Gateway also enables you to release APIs in the marketplace for other developers to purchase and use.

- API Gateway provides multiple security mechanisms to secure APIs and reduce the risks introduced by open APIs. These mechanisms include protection against replay attacks, request encryption, identity authentication, permission management, and throttling.
- API Gateway provides API lifecycle management that allows you to create, test, publish, and unpublish APIs. It also generates SDKs and API documentation to improve API management and iteration efficiency.

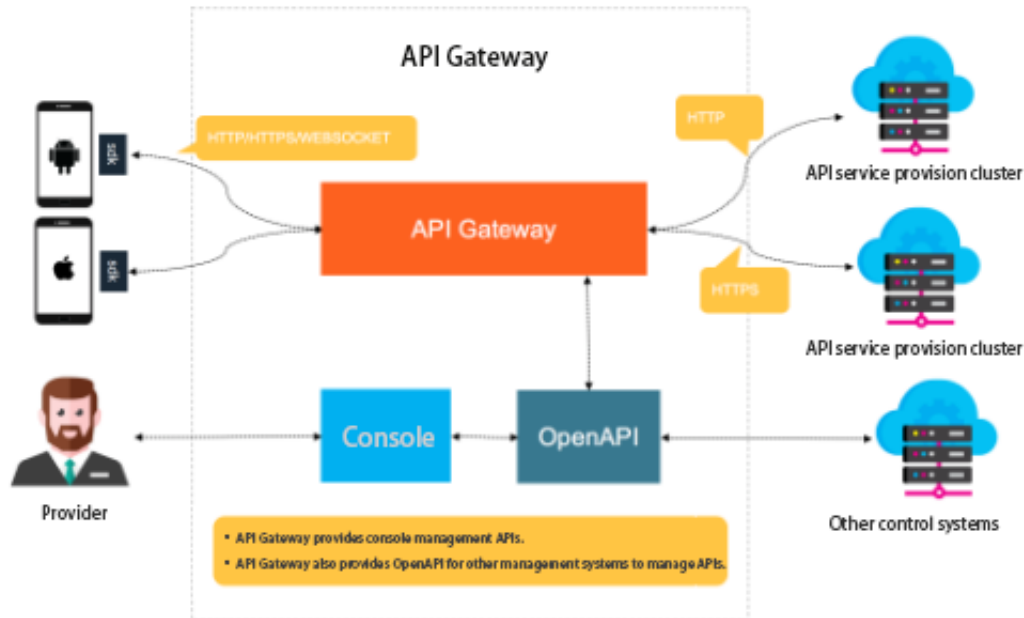
API Gateway allows enterprises to reuse and share their capabilities with each other so that they can focus on their core business.

Figure 21-1: API Gateway



## 21.2 System architecture

Figure 21-2: System architecture of API Gateway



API Gateway consists of the following components:

- Gateway

The gateway component is the core system that implements all business logic.

The gateway component supports multi-protocol access for all clients, 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 back-end APIs, and returns back-end responses to clients.

- OpenAPI

The OpenAPI component consists of a group of standard management APIs provided by API Gateway to manage API definitions. You can use the OpenAPI component to manage groups, metadata, and authorization for APIs. When the OpenAPI component receives an API change request, it synchronizes the change

to all gateway services. System administrators can use the management APIs to manage the APIs that are running in the 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 APIs in their own management systems to manage their own APIs.

- Console

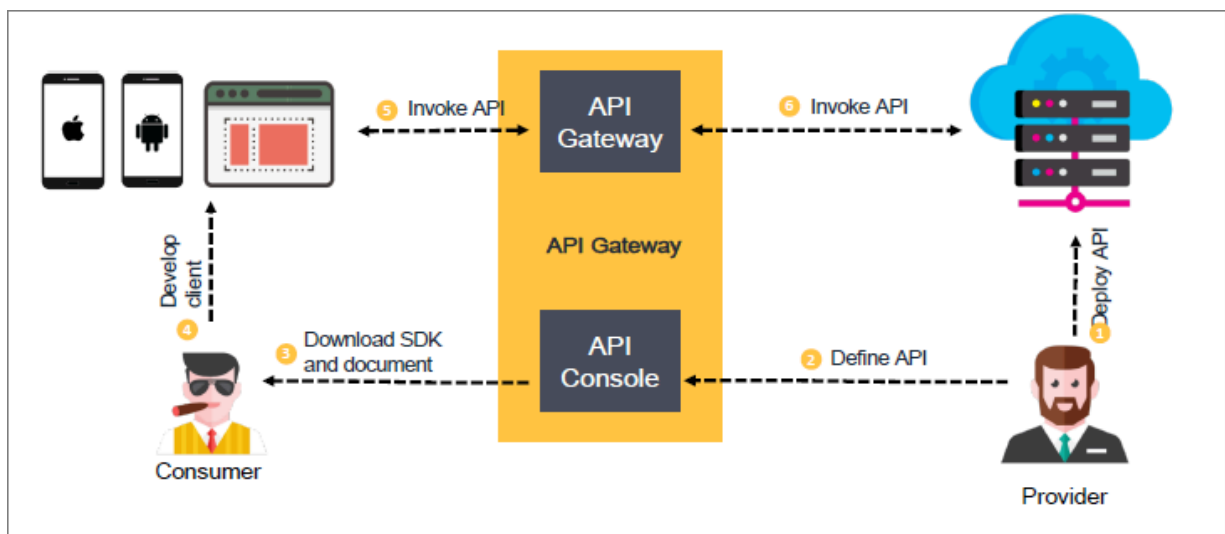
The API Gateway console implements all features of API Gateway. System administrators can manage their own APIs in the console in real time.

The API Gateway console provides you with a graphical user interface to call APIs through the OpenAPI component.

## 21.3 Features

### 21.3.1 API lifecycle management

Figure 21-3: API lifecycle management

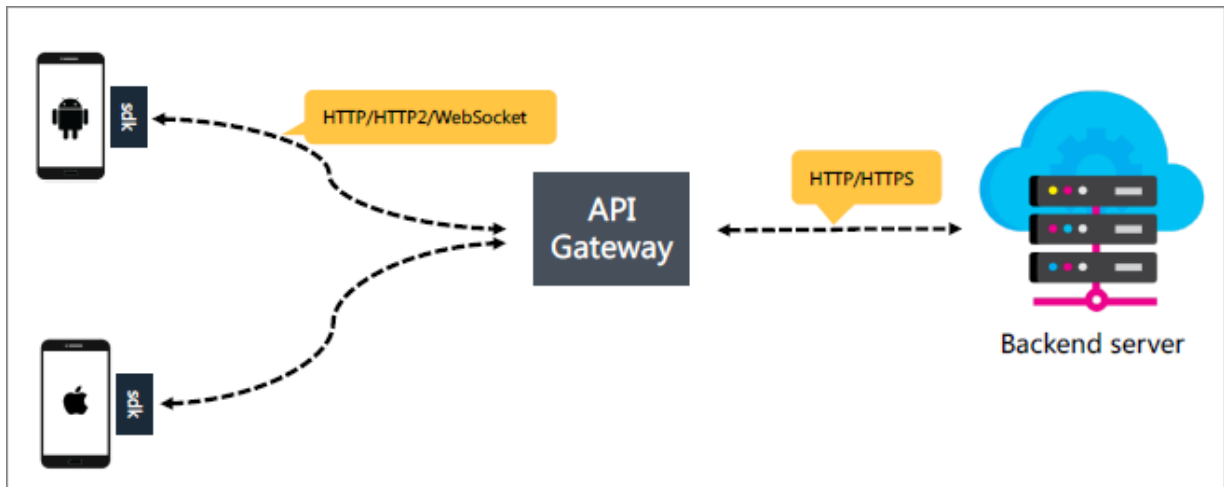


It provides a range of lifecycle management functions to publish, test, and unpublish APIs.

It provides maintenance functions such as routine API management, API version management, and quick API rollback.

## 21.3.2 Multi-protocol access

Figure 21-4: Multi-protocol access



The following protocols can be selected to establish bidirectional communication between clients and API Gateway:

- **HTTP:** is the most popular Internet text protocol.
- **HTTP/2:** supports multiplexing and header compression for high efficiency.
- **WebSocket:** supports persistent connections for binary communications.

Unlike HTTP/1.x, all HTTP/2 communication is split into smaller messages and frames, each of which is encapsulated with binary encoding. In HTTP/1.x, the header information is encapsulated in the Headers frame, and the request body is encapsulated in the Data frame. A single TCP connection can be used to send multiple requests. This reduces connections to the server and improves throughput. HTTP/2 uses header compression to enable faster data transmission and deliver more benefits to the mobile network environment so that network congestion is reduced.

Browsers limit the number of HTTP/1.x connections with the same domain name at the same time. If the limit is exceeded, further requests will be blocked. The binary framing layer in HTTP/2 enables full request and response multiplexing within a shared TCP connection. An HTTP message is divided into independent frames that are interleaved and reassembled on the other end based on stream identifiers and

headers. The following figures compare data transmission between HTTP/1.x and HTTP/2.

Figure 21-5: HTTP/1.x

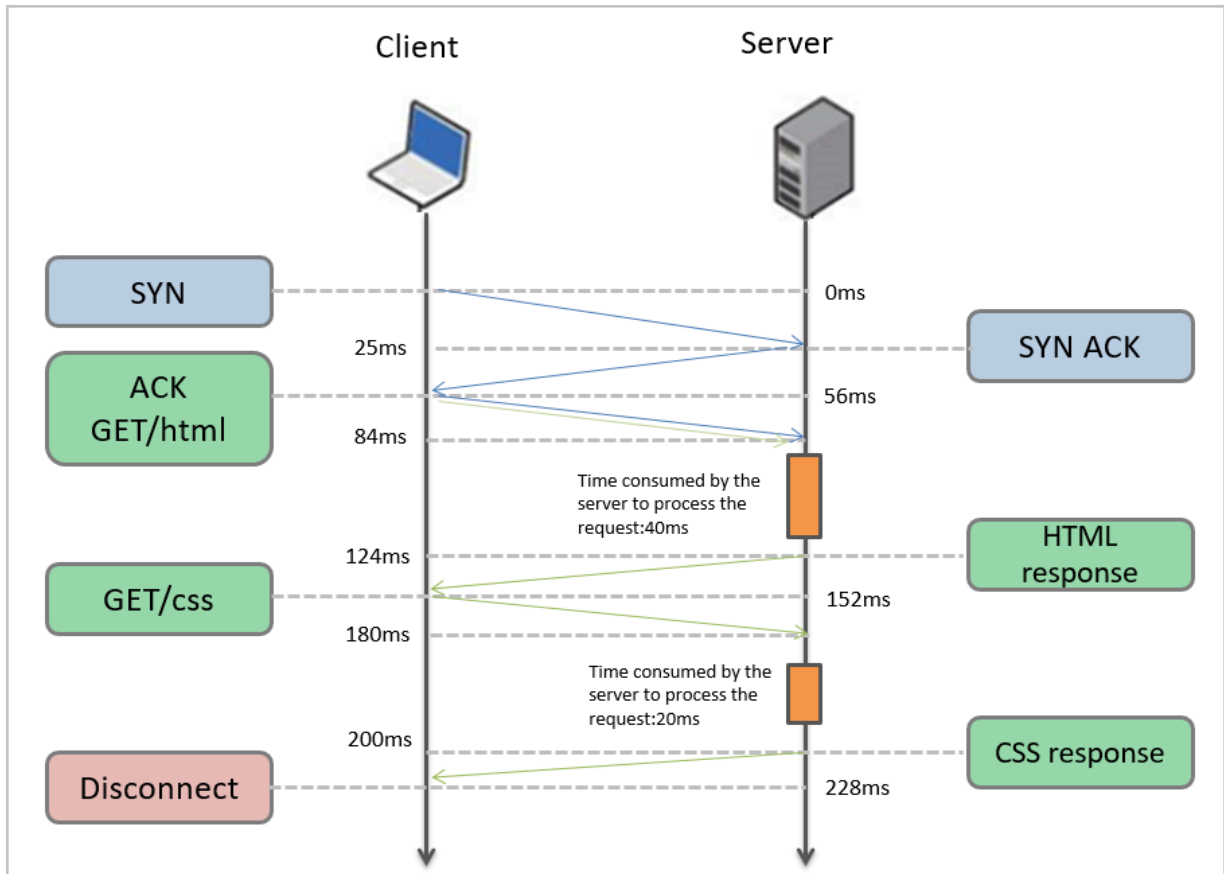
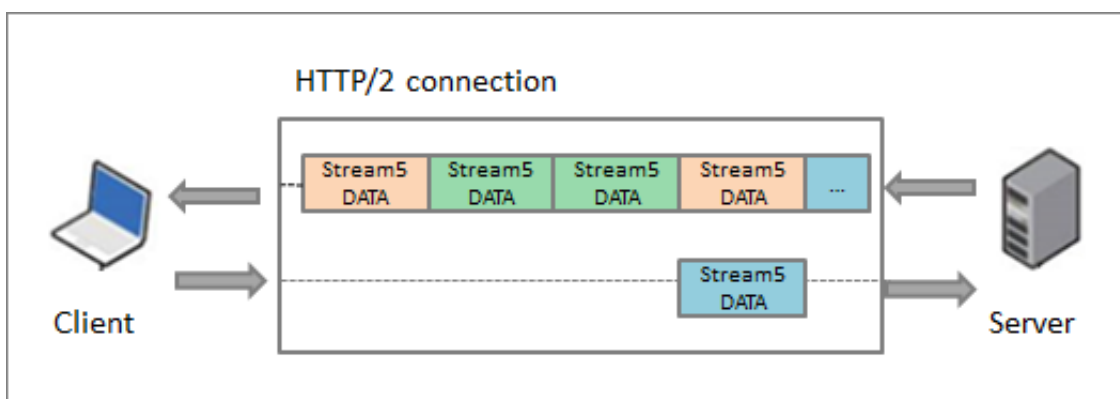
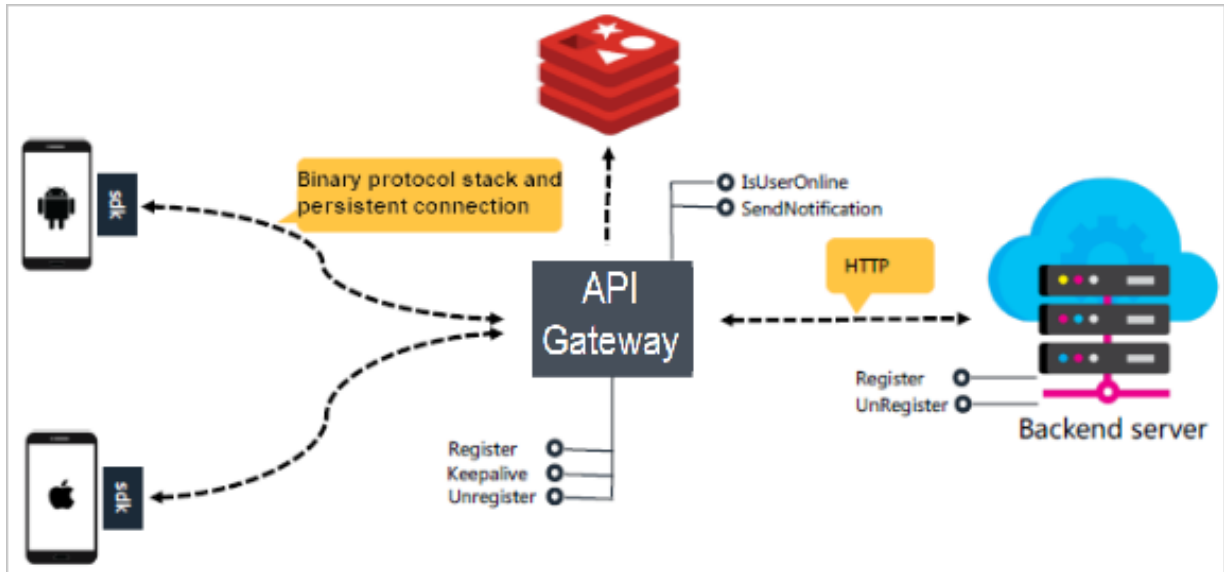


Figure 21-6: HTTP/2



**WebSocket** is a protocol that enables full-duplex persistent communication. Both clients and servers can send and receive data to and from each other. HTTP is used during the handshake. After a successful handshake, clients and servers can directly communicate with each other without HTTP. The data format is

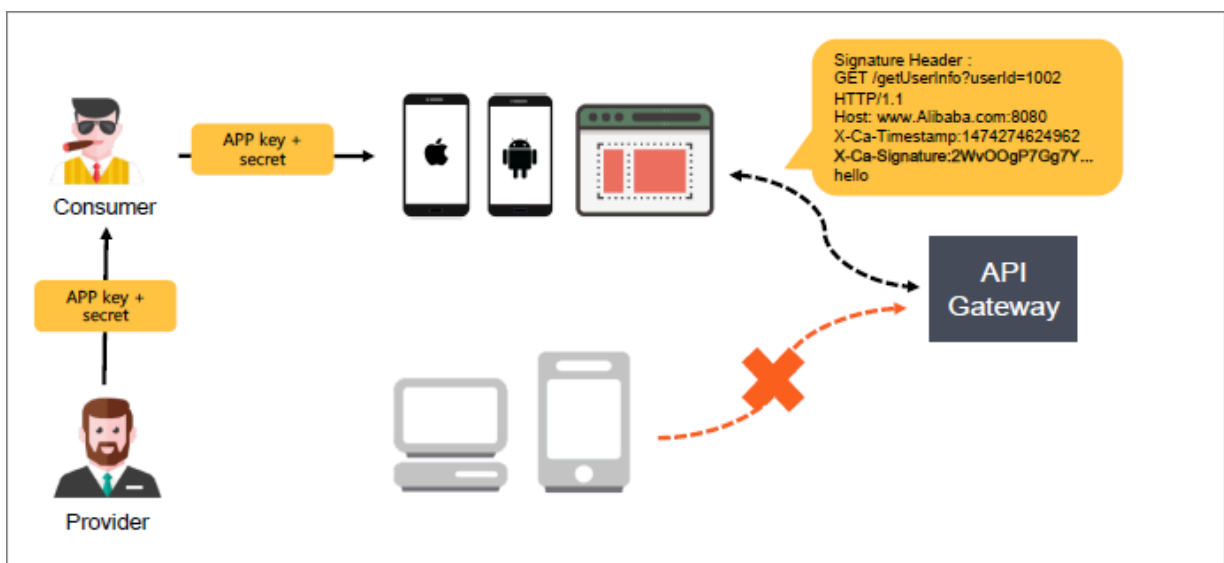
lightweight, the performance overheads are low, and the communication efficiency is high. Clients can communicate with any servers without the same-origin policy.



API Gateway supports bidirectional communication and maintains persistent connections between clients and itself. API Gateway can update the online status of clients after receiving heartbeat requests. Back-end services can access the API Gateway interface to query the online status of clients and push in-application notifications. API Gateway implements bidirectional communication based on the WebSocket protocol. Bidirectional communication is supported by Android SDKs, Objective-C SDKs, and Java SDKs.

### 21.3.3 Application access control

Figure 21-7: Application access control



API Gateway provides an application-based authentication mechanism. This mechanism ensures that only authorized clients can send requests to the back-end services. Applications are the identities that you use to call APIs. Each application has a key pair that consists of an AppKey and an AppSecret. The AppKey parameter is added to the request header when a client sends a request, while the AppSecret is used to calculate the signature. Signature verification can protect user data from being tampered. If the verification fails, an error is reported immediately.

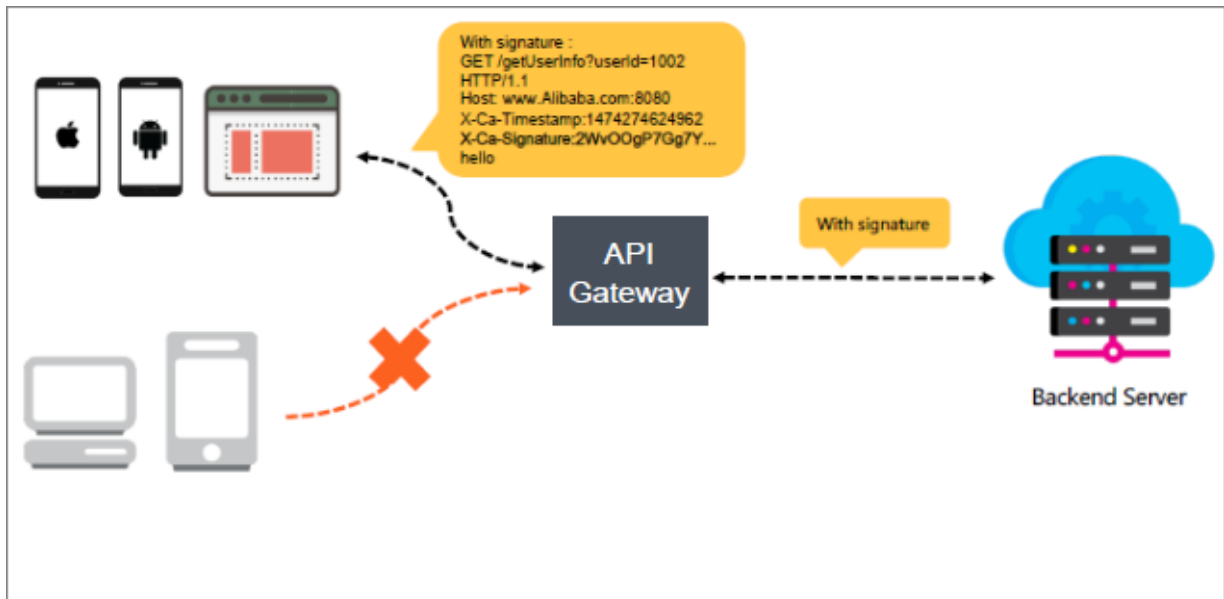
Symmetric encryption is used to verify the signature. The construction rules of the signature string are described in public documentation. HMAC-SHA256 is used as the signature algorithm. AppSecret is used as the encryption key that is only available to the clients and API Gateway. The client constructs and encrypts StringToSign based on specific rules. For more information about the construction methods, see Request signatures. After receiving the request, API Gateway constructs the StringToSign based on request parameters. Then, API Gateway finds the corresponding AppSecret through the AppKey. API Gateway calculates the signature string and compares it with the signature string that is sent by the user. If both signature strings are same, the signature verification is passed. Otherwise, it fails.

The principle of application-based authentication is as follows: API Gateway obtains the unique AppID based on the AppKey, and checks whether the application is authorized to access the API based on the AppID and API. If yes, access to the API is allowed. Otherwise, an unauthorized access error is reported.



### 21.3.4 Full-link signature verification mechanism

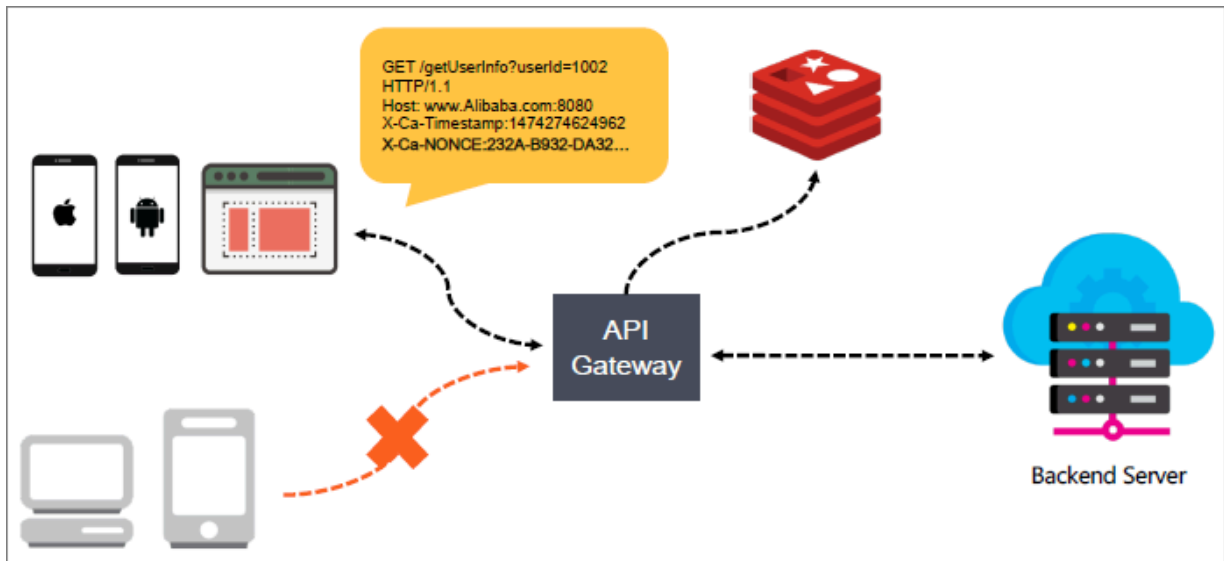
Figure 21-8: Full-link signature verification mechanism



API Gateway provides a full-link signature verification mechanism for communication between the client and API Gateway or between API Gateway and the backend service. This mechanism prevents data tampering during request transmission. When a client calls an API, the client must convert the key request data into a signature string based on API Gateway signature algorithms. The client must attach the signature string to the request header. API Gateway performs symmetric calculation to parse the signature and verify the identity of the request sender. HTTP, HTTPS, and WebSocket requests must have a signature in their header.

## 21.3.5 Anti-replay mechanism

Figure 21-9: Anti-replay mechanism



API Gateway provides an anti-replay mechanism to protect against data tampering used in replay attacks.

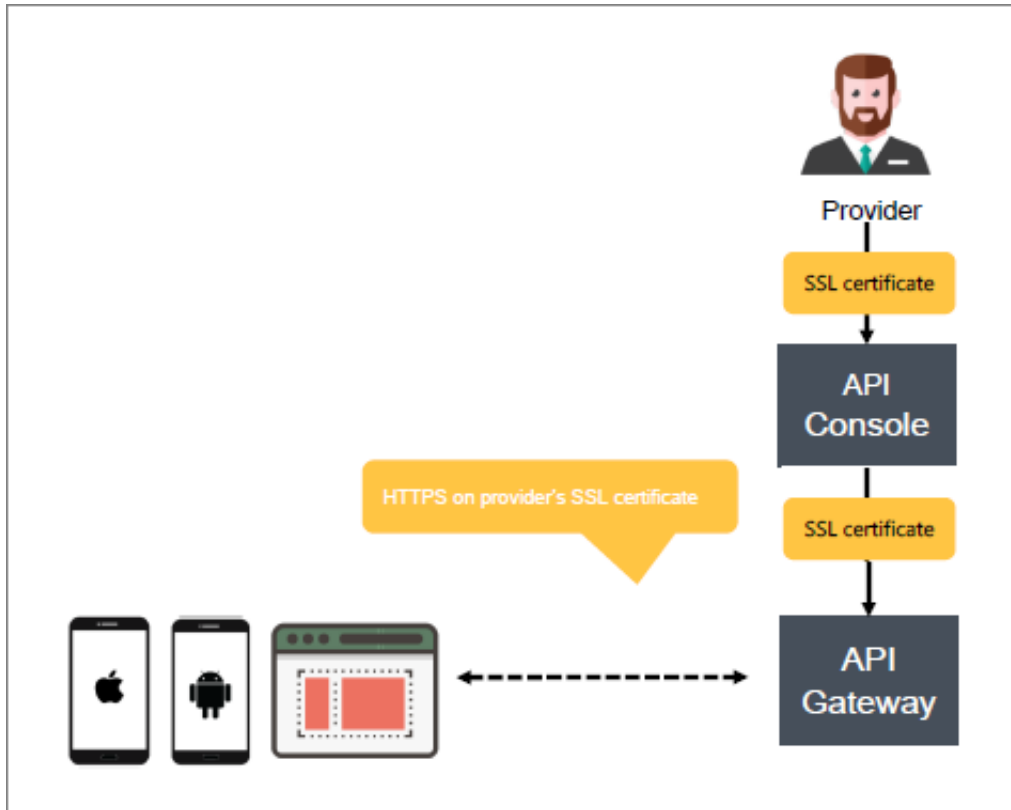
- When a client sends a request to API Gateway, the X-Ca-Nonce header is added. The value of the X-Ca-Nonce header can be any string. API Gateway verifies whether the same X-Ca-Nonce header has been passed in within 15 minutes. If yes, the request is considered a replay, and API Gateway reports an error immediately.

A distributed cache is used. API Gateway verifies whether the same X-Ca-Nonce header exists for each request.

- The value of the Nonce parameter is included in the signature string. Therefore, it cannot be tampered.

### 21.3.6 HTTPS communication based on the SSL certificate of the user

Figure 21-10: HTTPS communication based on the SSL certificate of the user

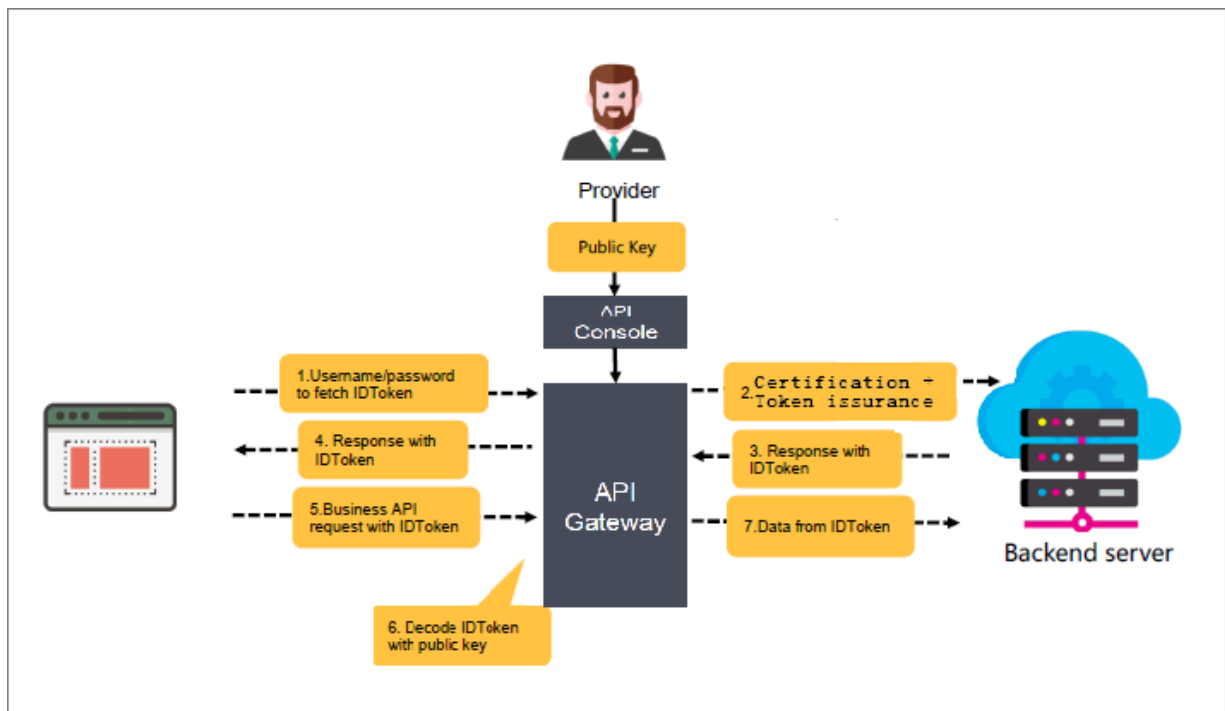


A system administrator can upload an SSL certificate corresponding to the domain name in the API Gateway console. Data transmitted between clients and API Gateway will then be encrypted based on the certificate. This prevents data tampering during transmission.

System administrators can update SSL certificates in real time in the API Gateway console.

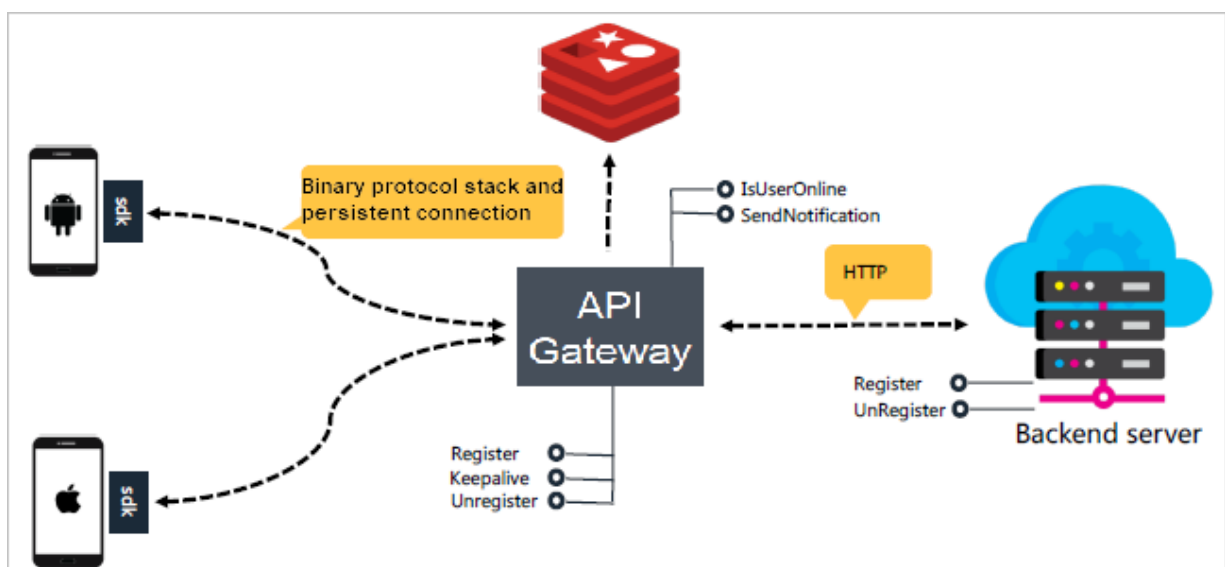
### 21.3.7 Support for OpenID Connect

Figure 21-11: Support for OpenID Connect



API Gateway supports OpenID Connect authentication, allowing API providers to verify requests based on their own user systems. OpenID Connect is a lightweight authentication standard based on OAuth 2.0. It provides a framework for identity interaction through APIs. Compared with OAuth, OpenID Connect not only authenticates a request, but also specifies the identity of the requester.

### 21.3.8 Bidirectional communication

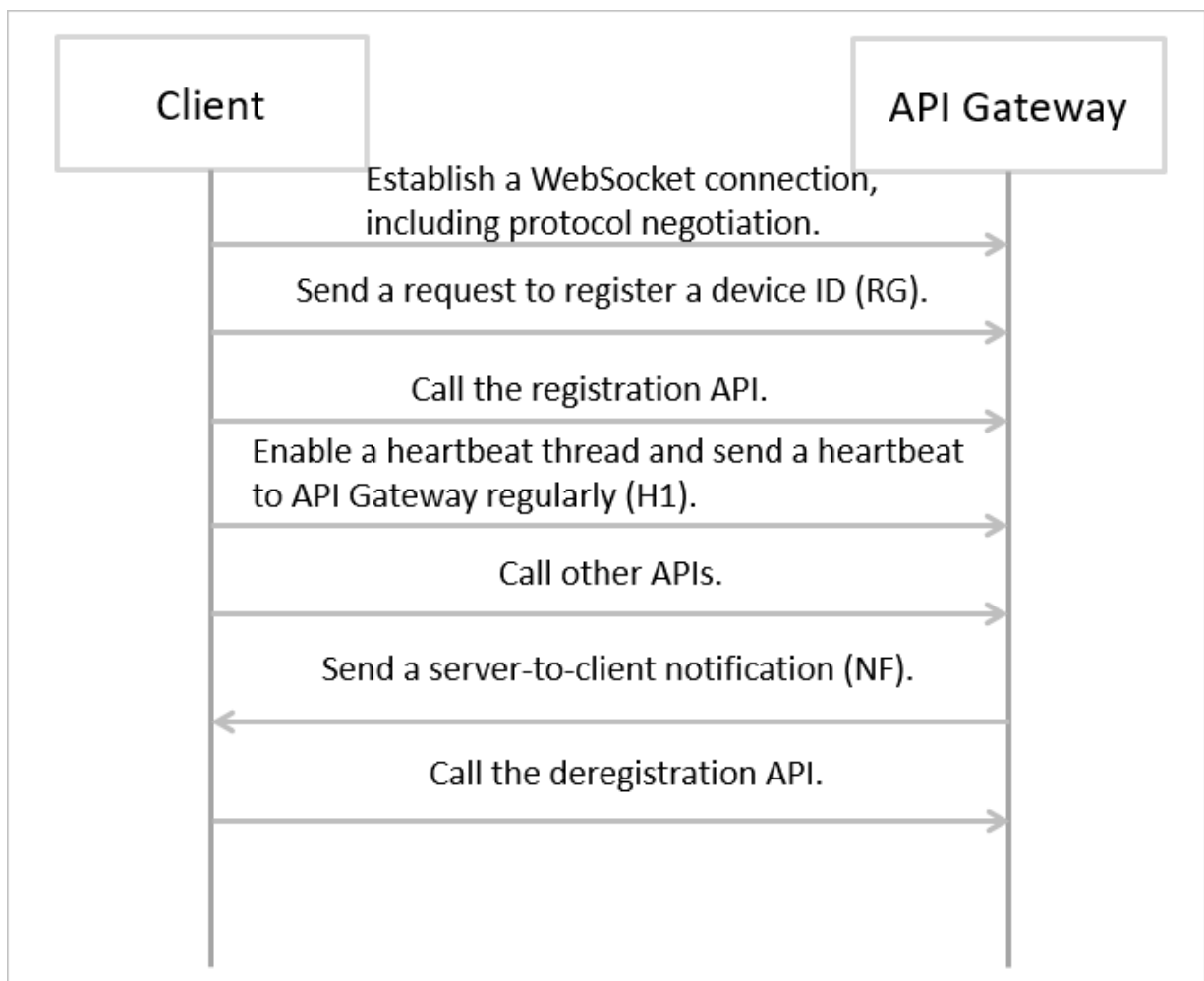


API Gateway supports bidirectional communication and maintains persistent connections between clients and itself. API Gateway can update the online status of clients after receiving heartbeat requests. Back-end services can access the API Gateway interface to query the online status of clients and push in-application notifications.

API Gateway implements bidirectional communication based on the WebSocket protocol. Bidirectional communication is supported by Android SDKs, Objective-C SDKs, and Java SDKs.

API Gateway provides built-in APIs to WebSocket users, including the APIs that clients use to register and deregister device IDs with API Gateway, and the APIs that are called to detect heartbeats.

Before establishing a WebSocket connection between clients and API Gateway, you must call the registration API to register device IDs. The following figure shows the interaction between clients and API Gateway.



Clients need to complete the following operations:

- 1. Establish a WebSocket connection, including protocol negotiation.**
- 2. Send a request to register a device ID (RG).**
- 3. Call the registration API.**
- 4. Enable a heartbeat thread and send a heartbeat to API Gateway regularly (H1).**
- 5. Call other APIs.**
- 6. Receive notifications sent by API Gateway.**
- 7. Call the deregistration API.**

## 21.3.9 Automatic generation of SDKs and API documentation

API Gateway can automatically generate Java, Objective-C, and Android SDKs for the APIs customized by providers. API Gateway can also generate API documentation. The following figure shows part of the API documentation.

Figure 21-12: Automatic generation of SDKs and API documentation

### API name: apitest

#### Description

#### Request information

HTTP protocol: HTTP

Call address: ca72233674a64a0d9778b397cdcb7025-cn-hangzhou.alicloudapi.com/api/test/[type]

Method: GET

#### Request parameters

Parameter	Location	Type	Required?	Description
headerParam	HEAD	STRING	false	
type	PATH	STRING	true	
queryParam	QUERY	STRING	true	

#### Response information

##### Response parameter type

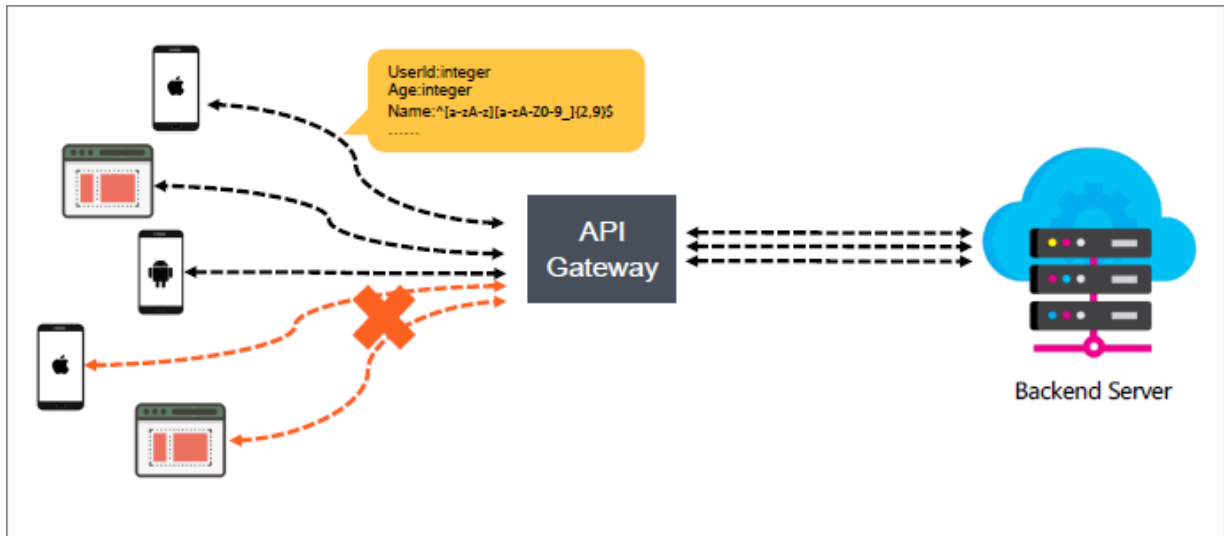
JSON

##### Returned result sample

```
test
```

### 21.3.10 Parameter cleaning

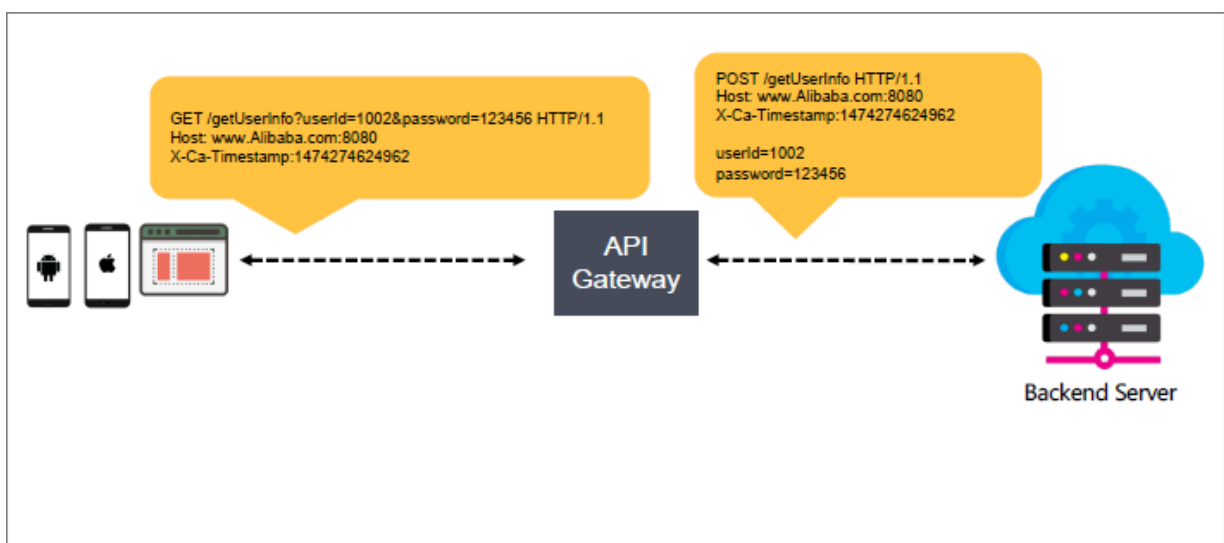
Figure 21-13: Parameter cleaning



System administrators can define the data type, regular expression, and enumeration of all API parameters. API Gateway forwards API requests that match the API definition to the backend service, while rejecting the requests that do not match the definition. This ensures that the backend service only receives standard requests that match API definitions.

### 21.3.11 Mappings between frontend and backend parameters

Figure 21-14: Mappings between frontend and backend parameters

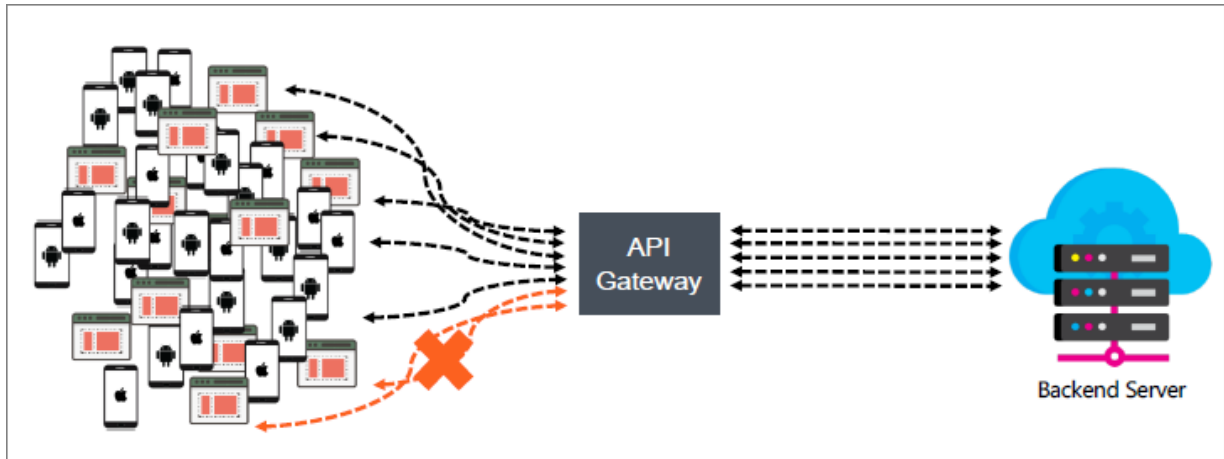


API Gateway provides parameter mapping capabilities to relocate parameters within a request before sending the request to the backend service. For example,



a parameter in a request sent to API Gateway is defined in Query. API Gateway can map the parameter to Form and then send the request to the backend service. This function ensures that users can access complicated backend functions by calling well-organized APIs.

### 21.3.12 Throttling

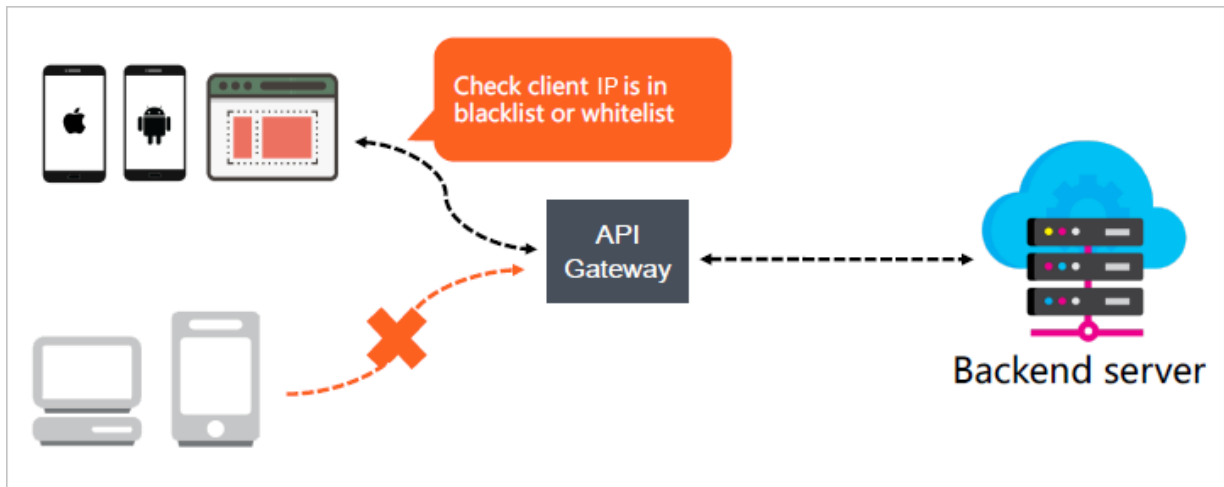


System administrators can set a request threshold based on the maximum processing capabilities of backend services. If the total number of requests exceeds the threshold, API Gateway will reject the excess requests to protect backend services from being overloaded. Supported dimensions include API, user, and application. Supported time granularity includes second, minute, hour, and day.

API Gateway uses a distributed cache, calculates the number of API requests from clients, and customizes keys in different formats based on the unit of time that you set. For example, if you set the unit of time to minute, the key is displayed in the yyyyMMddHHmm format. If the current time is 20:00 on May 7, 2019, the key is displayed as 201905072000. All requests in the current period are accumulated for this key. Requests will be recounted automatically during the next period. When another request is received, counting will be restarted.

### 21.3.13 IP address-based access control

Figure 21-15: IP address-based access control

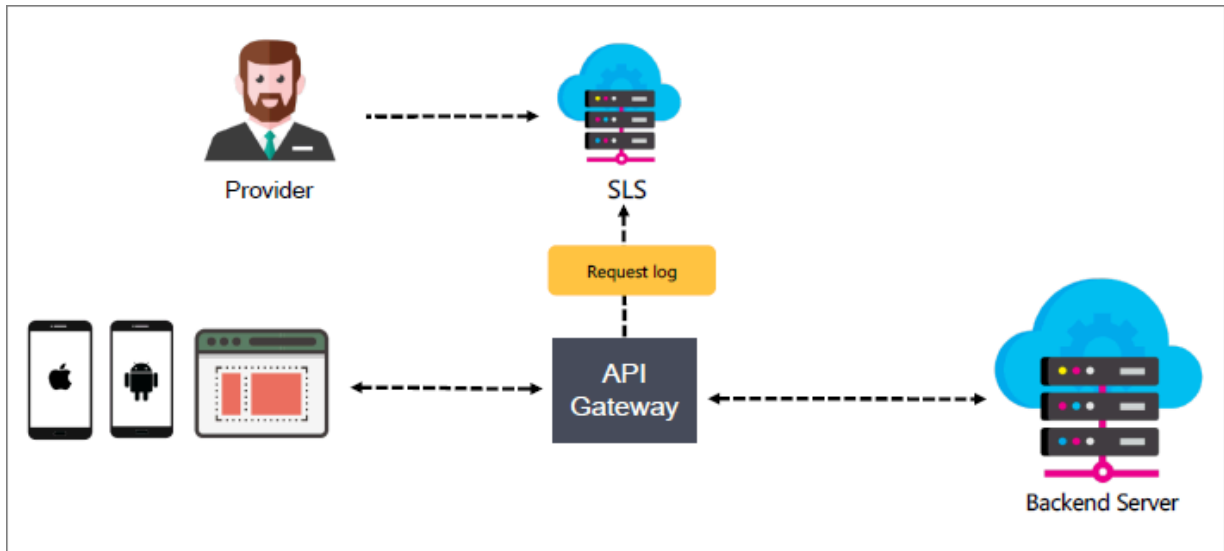


IP address-based access control is one of the API security protection measures provided by API Gateway. This measure controls the source IP addresses or IP address segments for API requests. System administrators can configure an IP whitelist or blacklist for an API to allow or deny an API request from an IP address.

The IP addresses obtained by API Gateway are egress IP addresses of clients. You cannot use the X-Forward-For header because its value can be randomly set by clients. API Gateway compares these client IP addresses with user-defined rules. It allows access to APIs from IP addresses in the whitelist and denies access to APIs from IP addresses in the blacklist.

### 21.3.14 Log analysis

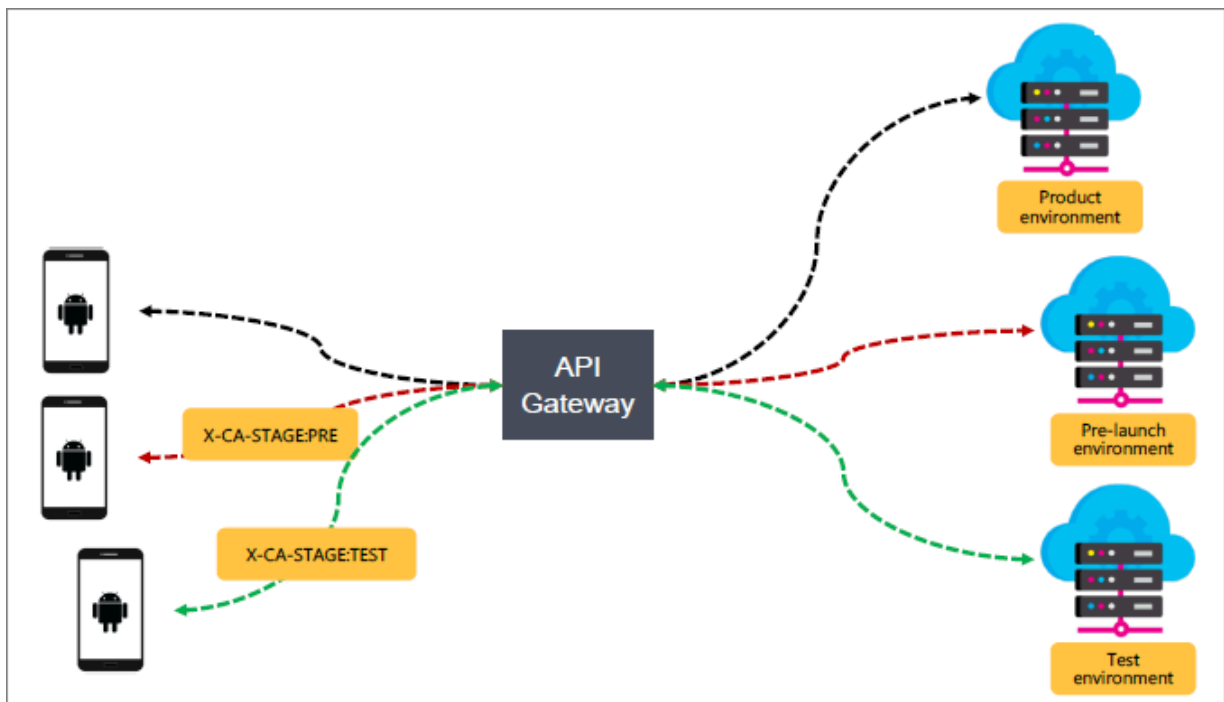
Figure 21-16: Log analysis



API Gateway sends called logs to Log Service. System administrators can use Log Service to query or download logs, or perform statistical analysis in real time. Logs can also be sent to OSS or MaxCompute.

### 21.3.15 Publish an API in multiple environments

Figure 21-17: Publish an API to multiple environments

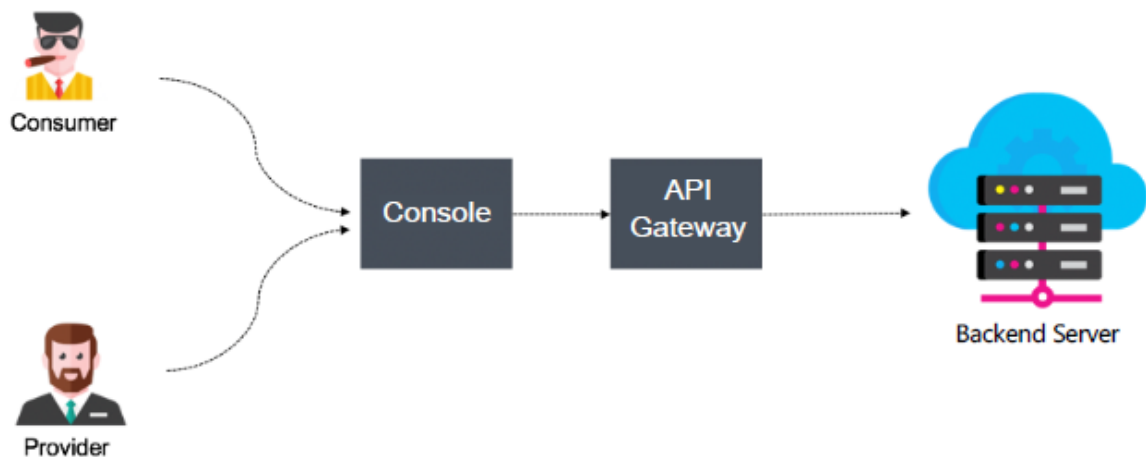


API Gateway allows you to publish an API group in three different environments: test, pre-release, and release environments. The test and pre-release environments are used by testers to test or debug APIs. The release environment is where the APIs can be used.

You can use the environment management function for API groups to set environment parameters for the test, pre-release, and online environments. The environment parameter is a common constant that can be customized for each environment. When you call an API, you can place the environment parameter in any location of the request. API Gateway identifies the environment based on the environment parameter in your request.

### 21.3.16 Online debugging

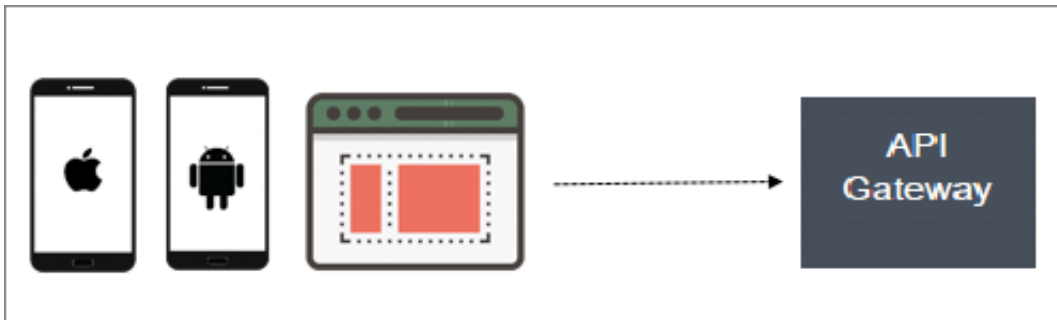
Figure 21-18: Online debugging



API Gateway provides the online API debugging function for system administrators and client developers.

### 21.3.17 Mock mode

Figure 21-19: Mock mode

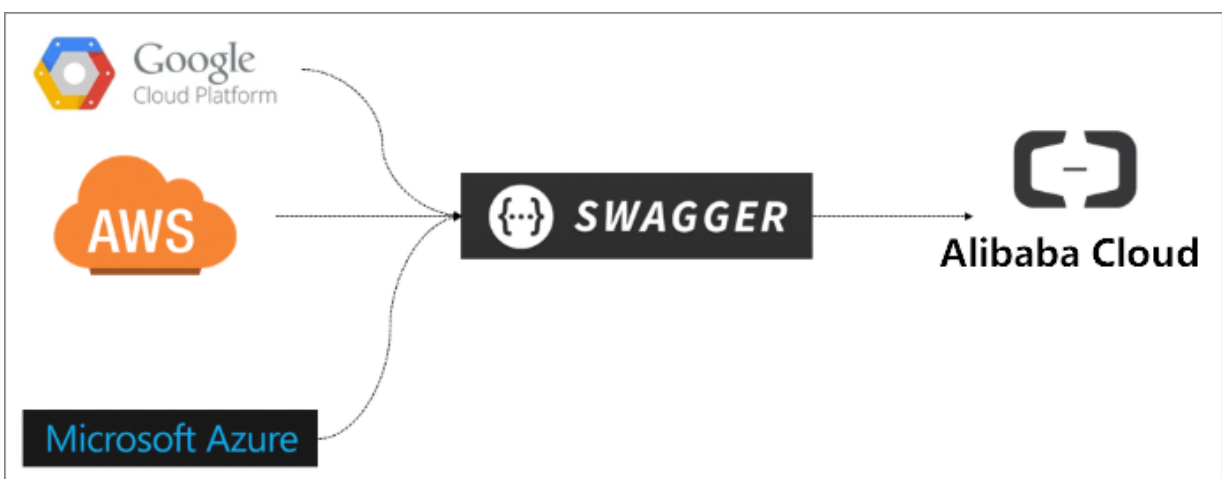


A project is typically developed by multiple partners working together toward a specific goal. The interdependence among various stakeholders often restricts individual members during the process, and misunderstandings may affect the development process or even delay the project schedule. You can mock expected responses to be returned to API callers during the project development process. This can greatly reduce misunderstanding among partners and greatly improve the development efficiency.

API Gateway supports the Mock mode.

### 21.3.18 Swagger file import

Figure 21-20: Swagger file import



Swagger is a widely-used specification to define and describe backend service APIs. You can create APIs in the API Gateway console by importing Swagger 2.0 files.

The API Gateway Swagger extension is based on Swagger 2.0. You can create the Swagger definition for API entities, and import the Swagger file to API Gateway for bulk creation or updating of API entities. API Gateway supports Swagger 2.0 by default, which is compatible with most Swagger specifications.

## 21.4 Benefits

- **Easy maintenance**

After you register APIs in API Gateway, API Gateway handles all the API management issues such as documentation maintenance, version management of APIs, and SDK maintenance. This significantly reduces routine maintenance costs.

- **High performance**

API Gateway maintains persistent connections between clients and API Gateway itself by supporting HTTP/2 and WebSocket. Both HTTP/2 and WebSocket are efficient binary protocols.

API Gateway uses a distributed deployment and scales out automatically to handle a large number of API requests with low latency. API Gateway offers reliable and efficient features for your back-end services.

- **Stability**

API Gateway has provided services to Alibaba Cloud public cloud users for over two years, and has a proven track record of performance. API Gateway provides stable services even in uncommon cases such as when over-sized packets are received, or when back-end services are unstable and slow to respond.

- **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.

API Gateway enables strict authorization management, anti-replay mechanism, parameter cleaning, IP address-based access control, and precise throttling. This ensures secure, stable and controllable services.

## 22 Enterprise Distributed Application Service (EDAS)

---

### 22.1 What is EDAS?

**Enterprise Distributed Application Service (EDAS) is a PaaS platform for application hosting and microservice management, providing full-stack solutions such as application development, deployment, monitoring, and O&M. It supports Dubbo, Spring Cloud, and other microservice runtime environments, helping you easily migrate applications to the cloud.**

Diverse application hosting environments

**You can select instance-exclusive ECS clusters, Container Service Kubernetes clusters, and user-created Kubernetes clusters based on your application systems and resource needs.**

Abundant microservice frameworks

**You can develop applications and services in the native Dubbo, native Spring Cloud, and HSF frameworks, and host the developed applications and services to EDAS.**

- **You can host Dubbo and Spring Cloud applications to EDAS by adding dependencies and modifying a few configurations. You have access to the functions of EDAS, such as enterprise-level application hosting, service governance, monitoring and alarms, and application diagnosis, without having to build ZooKeeper, Eureka, and Consul. This lowers the costs of deployment and O&M.**
- **HSF is the distributed RPC framework that is widely used within the Alibaba Group. It interconnects different service systems and decouples inter-system implementation dependencies. HSF unifies the service publishing and call methods for distributed applications to help you conveniently and quickly develop distributed applications. HSF provides or uses common function modules, and frees developers from various complex technical details involved in distributed architectures, such as remote communication, serialization, performance loss, and the implementation of synchronous and asynchronous calls.**

## Complete application management

**You can perform end-to-end management, service governance, and microservice management for your applications in the EDAS console.**

- **Application lifecycle management**

**EDAS provides end-to-end application management, allowing you to deploy, scale out, scale in, stop, and delete applications. Applications of all sizes can be managed in the EDAS console.**

- **Service governance**

**EDAS integrates a wide variety of service governance components, such as auto scaling, throttling and degradation, and health check, to deal with unexpected traffic spikes and crashes caused by dependencies. This greatly improves platform stability.**

- **Microservice management**

**EDAS provides the service topology, service statistics, and trace query functions to help you manage every component and service in a distributed system.**

## Comprehensive monitoring and diagnosis

**You can monitor the status of resources and services in applications in the EDAS console to promptly identify problems and quickly locate their causes through the logging and diagnosis components.**

- **Application monitoring**

**EDAS monitors the health status of application resources at the IaaS layer in real time, helping you quickly locate problems.**

- **Application diagnosis**

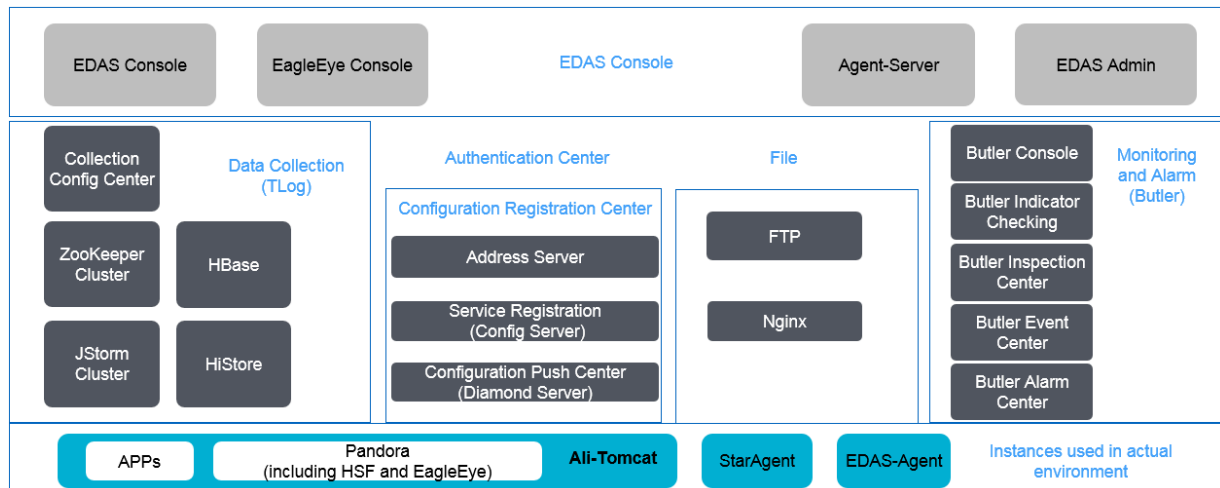
**EDAS provides the container-based application diagnosis function. Based on the provided data, this function allows you to identify application runtime errors , such as errors in Garbage Collection (GC), class loading, connectors, memory allocated for objects, thread hotspots, Druid database connection pools, and Commons Pool.**



## 22.2 Architecture

EDAS consists of the console, data collection system, configuration registry, and authentication center. *Figure 22-1: EDAS architecture* shows the EDAS architecture.

Figure 22-1: EDAS architecture



- **EDAS console**

It is a GUI where you can directly use EDAS system functions. In the console, you can implement resource management, application lifecycle management, O&M control, service governance, three-dimensional monitoring, and digital operations.

- **Data collection system**

It collects trace logs and the runtime statuses of EDAS clusters and all customer application instances, and summarizes, computes, and stores data in real time.

- **Configuration registry**

It is a central server used to publish and subscribe to HSF services (RPC framework) and push distributed configurations.

- **Authentication center**

It controls permissions for user data to ensure data security.

- **O&M system**

It is a major tool of EDAS for daily monitoring and alarms of all EDAS components.

- **Command channel system**

**It is a control center that remotely sends commands to application instances.**

- **File system**

**It stores WAR packages and required components, such as JDK and Ali-Tomcat, uploaded by users.**

## 22.3 Features and principles

### 22.3.1 Full compatibility with Apache Tomcat containers

**As the basic container for running EDAS applications, EDAS containers integrate with the Alibaba middleware technology stack to greatly improve the startup, monitoring, stability, and performance of containers. Also, EDAS containers are fully compatible with Apache Tomcat.**

### 22.3.2 Application-centric PaaS platform

Application management and O&M

**In the visual EDAS console, you can perform application lifecycle management tasks in a single place, including creating, deploying, starting, stopping, scaling out, scaling in, and deleting applications, thereby implementing full-process application management. With Alibaba's rich experience in O&M of ultra-large scale clusters, EDAS allows you to easily operate and maintain an application that has thousands of instances.**

Auto scaling

**EDAS supports scaling out and in applications both manually and automatically. With real-time monitoring of CPU, memory, and workload, you can scale out and in your applications in seconds.**

Primary account and RAM users

**EDAS provides a unique primary Alibaba Cloud account and RAM user system that allows you to establish primary account and RAM user relationships on the EDAS platform based on the organization of your enterprise's departments, teams, and projects. In addition, ECS resources are organized based on primary account and RAM user relationships to simplify resource allocation.**

## Role and permission control

**The maintenance of applications normally involves application development owners, application maintenance owners, and instance owners. Considering that different roles perform different management activities on an application, EDAS provides a role and permission control mechanism that allows you to define roles and assign permissions for different accounts.**

## 22.3.3 Rich distributed services

### Distributed service framework

**Since 2007, as the e-commerce platforms of the Alibaba Group continuously developed to distributed architectures, the self-developed distributed service framework High Speed Framework (HSF) and Dubbo came into being. Built on a high-performance network communication framework, HSF is a distributed service framework for enterprise Internet architectures. It provides proven features, such as service publishing, registration, calling, routing, authentication, throttling, degradation, and distributed tracing.**

### Distributed configuration management

**The transformation from a centralized system to a distributed system makes it a challenge to manage the configuration information on each instance of the distributed system in real time. EDAS provides efficient distributed configuration management that allows you to centrally manage all configuration information across the distributed system in the EDAS console. More importantly, EDAS allows you to modify the configuration information in the console and notifies all the instances of this modification in seconds.**

### Distributed task scheduling

**SchedulerX, a task scheduling service integrated in EDAS, allows you to configure any single-instance or distributed tasks for periodic scheduling. It also provides the ability to manage the running periods and query the running history of the tasks. It applies to task scheduling scenarios such as migrating historical data at two o'clock every morning, triggering a task every five minutes, or sending a monthly report on the first day of each month.**

## 22.3.4 Maintenance management and service governance

### Service authentication

**HSF is designed to ensure the reliability and security of each distributed call. Strict authentication is implemented in every phase, from service registration and subscription to service calling.**

### Service throttling

**EDAS can apply a number of throttling rules on each application to control service traffic and ensure service functionality. EDAS supports configuring throttling rules by QPS and thread to ensure maximum stability at traffic peaks.**

### Service degradation

**Contrary to service throttling, service degradation pinpoints and blocks poor services that your application calls. This feature ensures the stable operation of your application and prevents the functionality of your application from being compromised by dependency on poor services. EDAS allows you to configure degradation rules by response time, which effectively blocks poor services at traffic peaks.**

## 22.3.5 Three-dimensional monitoring

### Distributed tracing

**EDAS EagleEye analyzes every service call, sent message, and access to the database within the distributed system, so you can precisely identify system bottlenecks and risks.**

### Service call monitoring

**EDAS can fully monitor the service calls made by your application in terms of the QPS, response time, and error rate of your services.**

### Infrastructure monitoring

**EDAS can thoroughly monitor the running status of your application in terms of basic metrics such as CPU, memory, workload, network, and disk.**

## 22.4 Performance features

Feature	Specifications
<b>Access request processing capability</b>	In simple calling scenarios, without taking into account the uncertain response time of service providers, remote procedure calls (RPCs) have a 4,000 QPS per CPU core for a 1 KB service request message. With linear scalability, a single service registry supports 20,000 QPS.
<b>Basic features</b>	It provides Java containers that integrate multiple pieces of Internet middleware and is fully compatible with Apache Tomcat.
	It implements application lifecycle management, including publishing, starting, stopping, and scaling out or in applications.
	It monitors basic hardware metrics.
	It monitors Java containers.
	It provides a comprehensive service authentication mechanism.
	It enables distributed RPCs and processes messages and transactions with multiple data sources.
	It enables logging, inspection, monitoring, and tracing for service links and system metrics.
	It pushes distributed system configurations.
<b>Reliability</b>	The data system uses multi-level cache and primary/secondary storage solutions.
<b>Scalability</b>	Service nodes can be continuously scaled out and in.
<b>Proven technology</b>	It is based on the highly available and high-performance distributed cluster technology products that have been used within the Alibaba Group for a long period of time. In addition, the EDAS team members have rich experience in this field.

## 23 MaxCompute

---

### 23.1 What is MaxCompute?

#### 23.1.1 Overview

**MaxCompute is an offline data processing service developed by Alibaba Cloud based on the Apsara system. It is capable of processing large volumes of data. MaxCompute can process terabytes or petabytes of data in scenarios that do not have high real-time processing requirements. MaxCompute is used in fields such as log analysis, machine learning, data warehousing, data mining, and business intelligence.**

**MaxCompute provides an easy-to-use approach to analyze and process large amounts of data without deep knowledge of distributed computing. MaxCompute is widely implemented by Alibaba across its businesses for tasks such as data warehousing and BI analysis for large Internet enterprises, website log analysis, e-commerce transaction analysis, and exploration of user characteristics and interests.**

**MaxCompute provides the following features:**

- **Data channel**
  - **Tunnel:** provides highly-concurrent offline upload and download services. The tunnel service enables you to upload or download large volumes of data to or from MaxCompute. You must use a Java API to access the tunnel service.
  - **DataHub:** provides real-time upload and download services. Unlike data uploaded through the tunnel service, data uploaded through DataHub is available immediately.
- **Computing and analysis**
  - **SQL:** MaxCompute stores data in tables, and provides SQL query capabilities to manipulate the data. MaxCompute can be used as database software capable of processing terabytes or petabytes of data. MaxCompute SQL does not support transactions, indexes, or operations such as UPDATE and DELETE. The SQL syntax used in MaxCompute is different from that in Oracle and MySQL. SQL statements from other database engines cannot be migrated

seamlessly to MaxCompute. MaxCompute SQL responds to queries within a few minutes or seconds, instead of milliseconds. MaxCompute SQL is easy to learn. You can get started with MaxCompute SQL based on your prior experience of database operations, without having a deep understanding of distributed computing.

- **MapReduce:** Initially proposed by Google, MapReduce is a distributed data processing model that has become popular and widely implemented for a variety of business scenarios. This topic briefly describes the MapReduce model. You must have a basic knowledge of distributed computing and relevant programming experience before using MapReduce. MapReduce provides a Java API.
- **Graph:** a processing framework designed for iterative graph computing. Graph computing jobs use graphs to build models. A graph is a collection of vertices and edges that have values. MaxCompute Graph iteratively edits and evolves graphs to obtain analysis results.
- **Unstructured data access and processing (integrated computing scenarios):** Alibaba Cloud introduced the MaxCompute-based unstructured data processing framework so that MaxCompute SQL commands can directly process external user data, such as unstructured data from OSS. You are no longer required to first import data into MaxCompute tables.

MaxCompute allows you to process the following data sources by creating external tables:

■ **Internal data sources:** OSS, Table Store, AnalyticDB, ApsaraDB for RDS, HDFS (Alibaba Cloud), and TDDL.

■ **External data sources:** HDFS (Open Source), ApsaraDB for MongoDB, and Hbase.

- **Unstructured data access and processing in MaxCompute:** By reading data from and writing data to volumes, MaxCompute can store unstructured data, which otherwise must be stored in an external storage system.
- **Spark on MaxCompute:** a big data analytics engine designed by Alibaba Cloud to provide big data processing capabilities for Alibaba, government agencies, and enterprises.

- **Elasticsearch on MaxCompute: an enterprise-class system to retrieve information from large volumes of data and provide near-real-time search performance for government agencies and enterprises.**

## 23.1.2 Features and benefits

### Features

- **MaxCompute is a distributed system designed for big data processing.**  
MaxCompute is a core service in the Alibaba Cloud computing solution that is used to store and compute structured data. It is also a basic computing component of the Alibaba Cloud big data platform. MaxCompute is designed to support multiple tenants and provide data security and horizontal scaling. Based on an abstract job processing framework, the service provides centralized programming interfaces for various data processing tasks of different users.
- **MaxCompute uses a distributed architecture that can be scaled as needed.**
- **MaxCompute provides an automatic storage and fault tolerance mechanism to ensure high data reliability.**
- **MaxCompute allows all computing tasks to run in sandboxes to ensure high data security.**
- **MaxCompute uses RESTful APIs to provide services.**
- **MaxCompute can upload or download high-concurrency, high-throughput data.**
- **MaxCompute supports two service models: the offline computing model and the machine learning model.**
- **MaxCompute supports data processing methods based on programming models such as SQL, MapReduce, Graph, and MPI.**
- **MaxCompute supports multiple tenants, allowing multiple users to collaborate on data analysis.**
- **MaxCompute provides user permission management based on ACLs and policies , allowing you to configure flexible data access control policies to prevent unauthorized access to data.**
- **MaxCompute provides Elasticsearch on MaxCompute for enhanced applications.**
- **MaxCompute provides Spark on MaxCompute for enhanced applications.**
- **MaxCompute supports access and processing of unstructured data.**



## Benefits

- **China's only big data cloud service and real data sharing platform:** Warehousing, mining, analysis, and sharing of data can all be performed on the same platform. Alibaba Group implements this unified data processing platform in several of its own products such as Aliloan, Data Cube, DMP (Alimama), and Yu'e Bao.
- **Support for large numbers of clusters, users, and concurrent jobs:** A single cluster can contain more than 10,000 servers and maintain 80% linear scalability. A single MaxCompute instance can support more than 1 million servers in multiple clusters without restrictions (linear scalability is slightly affected). It supports the local multi-IDC mode. It supports over 10,000 users, over 1,000 projects, and over 100 departments (of multi-tenants). It 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 have to worry about the storage difficulties and the prolonged computing time caused by the increasing data volume. MaxCompute automatically expands the storage and computing capabilities of clusters based on the volume of data to process, allowing you to focus on maximizing the efficiency of data analysis and mining.
- **Out-of-the-box service:** You do not have to worry about cluster construction, configuration, and O&M. Only a few simple steps are required to upload data, analyze data, and obtain analysis results in MaxCompute.
- **Secure and reliable data storage:** User data is protected against loss, theft, and exposure by the multi-level data storage and access security mechanisms. These mechanisms include multi-replica technology, read/write request authentication, and application and system sandboxes.
- **Multi-tenancy for multi-user collaboration:** You can have multiple data analysts in your organization to work together by configuring different data access policies, while ensuring that each analyst can only access data within their own permissions. This maximizes work efficiency while ensuring data security.

### 23.1.3 Benefits

Compared with traditional databases, MaxCompute has the following benefits.

Table 23-1: Comparison of benefits

Benefit	Traditional databases	MaxCompute
System scalability	Disks cannot be shared across more than 100 nodes. Table and database sharding causes application data collision, resulting in massive computing overhead. This significantly compromises application analysis capabilities.	MaxCompute supports more than 10,000 nodes that can store more than 1.5 EBs of data. For example, during Alibaba's Double 11 event, MaxCompute processed more than 300 PBs of data in six hours.
Data type support	Cannot process unstructured data.	Can process both structured and unstructured data.
High availability	Redundant storage solutions are not available. Traditional backup and recovery approaches are inapplicable to large volumes of data (measured in PBs), and a single point of failure can cause the entire database to become unavailable.	Provides the shared-nothing architecture and multi-replica data model. This eliminates single points of failure.
Complex computing capability	Iterative computing and graph computing capabilities are not available. The disk sharing technology and complex computing operations result in massive data exchanges between nodes, imposing tremendous bandwidth pressure.	Provides distributed storage and multiple computing frameworks such as MR, SQL, iterative computing, MPI, and graph computing.

Benefit	Traditional databases	MaxCompute
Concurrency	A single large-scale computing task (such as index computing) can consume all system resources, and incur network and disk (data dictionary) bottlenecks. This makes highly concurrent access impossible.	Provides comprehensive multi-tenant isolation and resource management tools, so that you can easily view cluster resources and manage the resources used by each service. It can support up to 10,000 concurrent access requests.
Performance support	The indexing mechanism makes it difficult to support analytical applications of real-time data. Large amounts of data collision cause analytical predictions to take more than 24 hours, resulting in a performance bottleneck.	Focuses on the concurrent computing of large amounts of data. It provides available real-time data, and multiple high-performance computing capabilities, such as high-performance large-scale offline computing, real-time multi-dimensional analysis of large amounts of data, and stream computing.

### 23.1.4 Scenarios

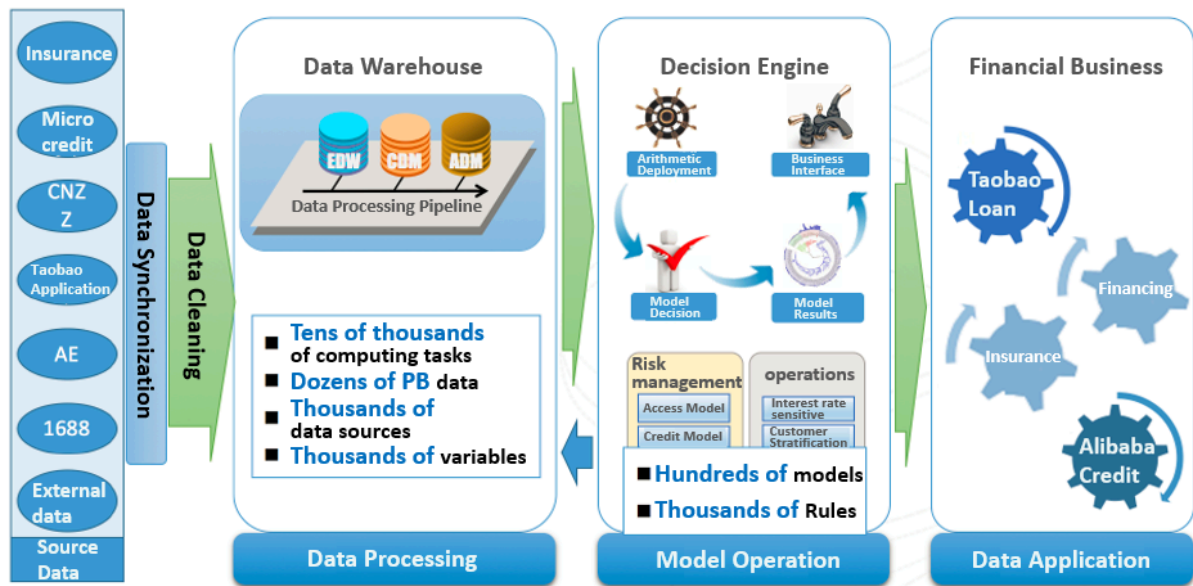
MaxCompute is designed for use in three big data processing scenarios:

- Establishment of SQL-based large data warehouses and BI systems
- Development of big data applications based on MapReduce and MPI distributed programming models
- Development of big data statistics models and data mining models based on statistics and machine learning algorithms

The following describe some real-world scenarios.

## Data warehouse construction

Figure 23-1: Data warehouse construction



MaxCompute enables you to easily build a cloud-based data warehouse. With MaxCompute capabilities such as partitioning, data table statistics, and table life cycle management, you can easily enhance the storage of historical data warehouse information, divide hot and cold tables, and control data quality.

Alibaba's financial data warehousing team has built a sophisticated and powerful data warehousing system based on MaxCompute. This system provides six layers: the source data layer, ODS layer, enterprise data warehousing layer, common dimensional modeling layer, application marketplace layer, and presentation layer.

- The source data layer processes data from all sources, including Taobao, Alipay, B2B, and external data sources.
- ODS provides a temporary storage layer for data import.
- The enterprise data warehousing layer uses the 3NF modeling technique to divide data, including all historical data, by topic (such as item or shop).
- The common dimensional modeling layer uses the dimensional modeling approach to create modeling layers for general business applications. This layer shields the upper layers from changes in business requirements, and provides consistent and actionable data to the upper layers.
- The application marketplace layer is a demand-oriented layer that provides a data marketplace for specific applications.

- The presentation layer provides several data portals and services that can be accessed by applications.

This system architecture inevitably involves tasks such as metadata management.

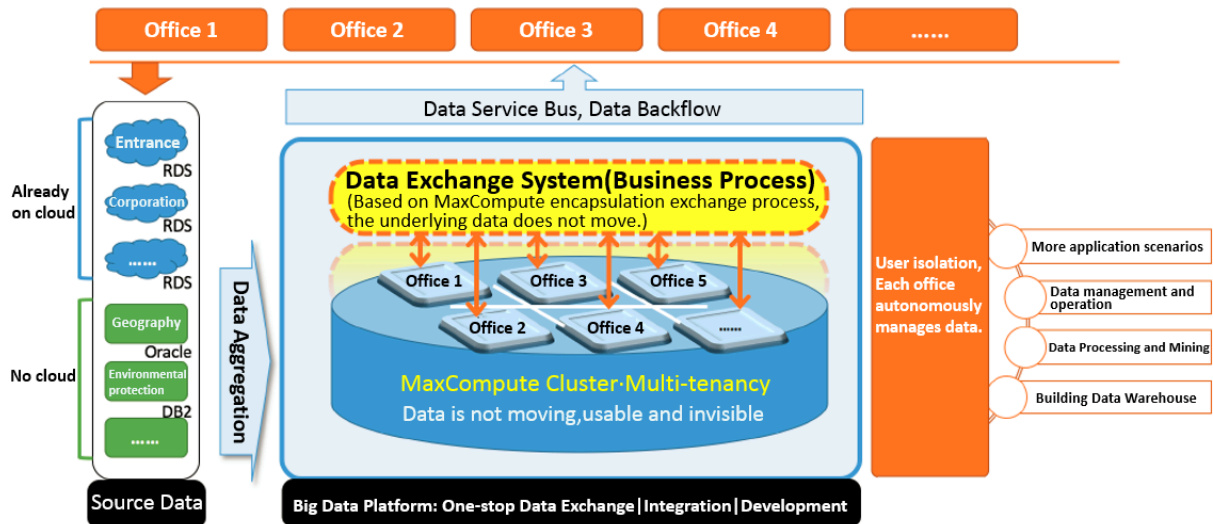
The financial data warehouse is used to perform offline computing tasks based on MaxCompute SQL. It also uses a series of metric rules and algorithms to make decisions offline for online decision-making.

MaxCompute-based data warehouses differ from traditional databases in the following ways:

- **Historical data storage:** MaxCompute is able to store large amounts of data. You do not have to dump historical data to cheaper storage media as you would do in traditional databases.
- **Partitioning:** Traditional databases provide a wide range of partitioning methods such as range partitioning. MaxCompute provides fewer partitioning methods, but are sufficient for use in data warehousing scenarios. Whatever the method, you can build a data warehouse based on the same concept and principle as a table partition.
- **Wide tables:** MaxCompute stores data in fields, making it ideal for creating wide tables.
- **Data integration:** Traditional databases use stored procedures for data processing and integration. MaxCompute splits the logic of these operations into discrete SQL statements. Though the implementation is different, the algorithms are the same. In many years of experience, we found that splitting the operation logic into discrete SQL statements is clearer and more efficient, while stored procedures are more flexible and capable of processing complex logic.

## Big data sharing and exchange

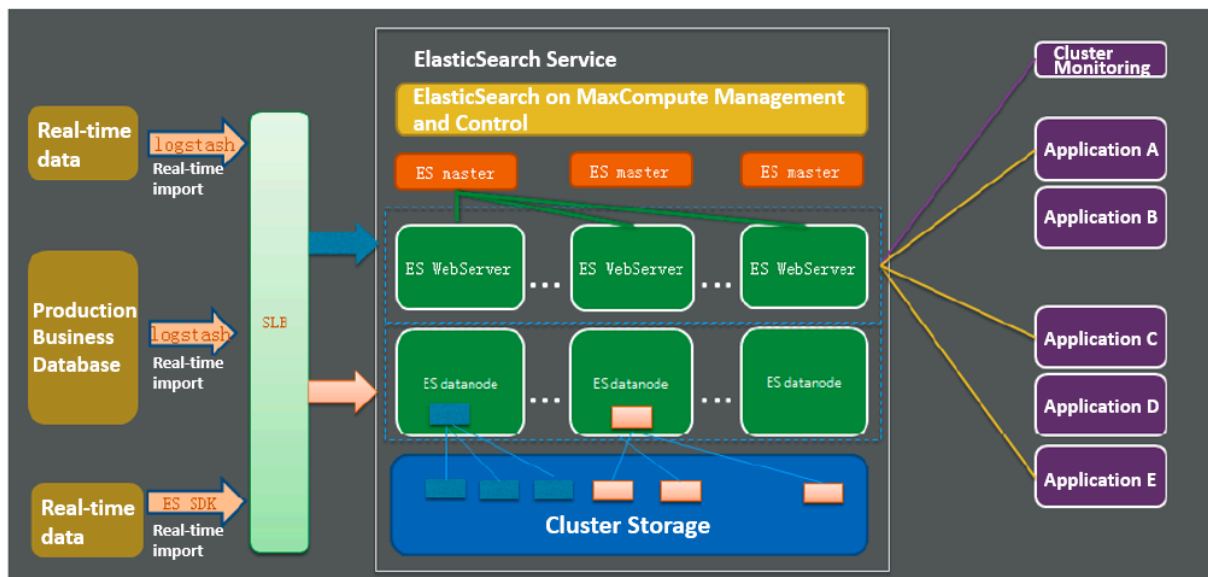
Figure 23-2: Big data sharing and exchange



MaxCompute provides a wide range of permission management methods and flexible data access control policies. MaxCompute provides a wide range of access control mechanisms, including the ACL authorization, role-based authorization, policy authorization, cross-project authorization, and label security mechanism. MaxCompute provides column-level security solutions. This can meet the security requirements within an organization or across multiple organizations. For projects that demand high security, MaxCompute provides the project protection mechanism to prevent data leakage, and provides logs of all user operations to facilitate retrospective audits.

## Typical applications of Elasticsearch on MaxCompute

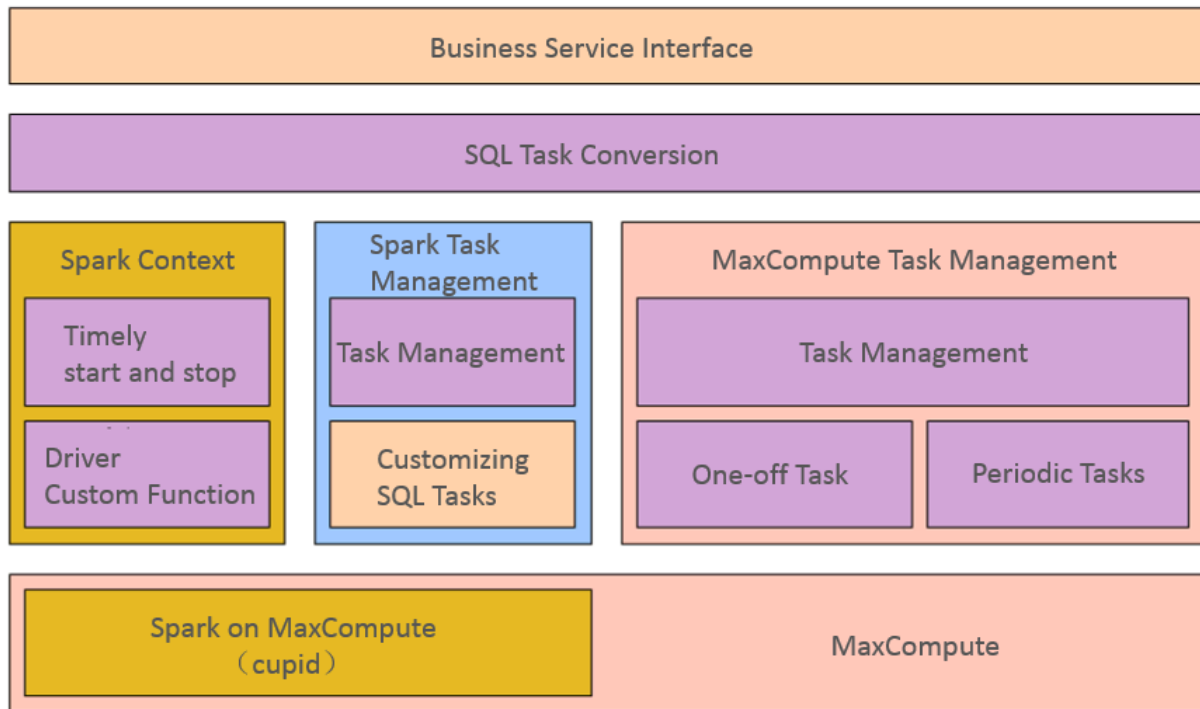
Figure 23-3: Typical applications



Elasticsearch on MaxCompute allows you to launch a set of Elasticsearch services by submitting jobs in a MaxCompute cluster. Native Elasticsearch code is not modified when applied in a project. Elasticsearch on MaxCompute runs in the same way as native Elasticsearch clusters.

## Typical applications of Spark on MaxCompute

Figure 23-4: Typical applications



Spark on MaxCompute provides business computing platform and applications in Client mode. The preceding figure shows the application framework.

## 23.1.5 Service specifications

### 23.1.5.1 Software specifications

#### 23.1.5.1.1 Overview

This section describes the software specifications of MaxCompute.

#### 23.1.5.1.2 Control and service

Table 23-2: Specifications

Item	Description
Number of control nodes	Greater than or equal to 3.
Number of MaxCompute front-end servers	Greater than or equal to 2. MaxCompute front-end servers can be deployed together with control nodes.
Number of tunnels	Greater than or equal to 2. Tunnels can be deployed together with compute nodes.



Item	Description
Number of DataHubs	Greater than or equal to 2. DataHubs can be deployed together with compute nodes.

### 23.1.5.1.3 Data storage

Table 23-3: Specifications

Item	Description
Logical storage capacity per node	12 TB
Total storage capacity	The storage capacity can be scaled out by adding more nodes.



**Note:**

The size of logically stored data to a large extent determines the size of the cluster to be evaluated.

### 23.1.5.1.4 Size of a single cluster

Table 23-4: Specifications

Item	Description
Offline computing cluster	An offline computing cluster can contain 3 to 10,000 machines.


### 23.1.5.1.5 Projects

Table 23-5: Specifications

Item	Description
Creation of projects	Supported.
Acquisition of project metadata	Supported.
Deletion of projects	Supported.
Setting of the default lifecycle of tables	Supported.
Number of supported projects	Over 1,000


## 23.1.5.1.6 User management and security and access control

Table 23-6: Specifications

Item	Description
Cross-project access	Supported. You can authorize cross-project access to organize tables and resources as packages and install them in other projects.
Service (odps_server and tunnel) authentication and access control	Supported. AccessKey ID and AccessKey Secret can be used to authenticate users and control their permissions.
Prevention of data outflow from a project	You can prevent data outflow and specify exceptions when necessary.
Label-based security	<p>Label-based security (LabelSecurity) can be set to enable column-level access control.</p> <div>  <b>Note:</b>            LabelSecurity is a mandatory access control policy that provides a wide range of security level settings.         </div>
Authorization to users	Supported.
Authorization to roles	Supported. You can customize roles and assign roles to users. Different roles are granted different permissions.

Item	Description
<b>Project-specific authorization</b>	<p>The following permissions can be granted on a project:</p> <ul style="list-style-type: none"> <li>• View project information (excluding any project objects), such as the creation time.</li> <li>• Update project information (excluding any project objects), such as comments.</li> <li>• View the list of all object types in the project.</li> <li>• Create tables in the project.</li> <li>• Create instances in the project.</li> <li>• Create functions in the project.</li> <li>• Create resources in the project.</li> <li>• Create volumes in the project.</li> <li>• Grant all preceding permissions.</li> </ul>
<b>Table-specific authorization</b>	<p>The following permissions can be granted on a table:</p> <ul style="list-style-type: none"> <li>• Read table metadata.</li> <li>• Read table data.</li> <li>• Modify table metadata.</li> <li>• Overwrite or add table data.</li> <li>• Delete the table.</li> <li>• Grant all preceding permissions.</li> </ul>
<b>Function-specific authorization</b>	<p>The following permissions can be granted on a function:</p> <ul style="list-style-type: none"> <li>• Read.</li> <li>• Update.</li> <li>• Delete.</li> <li>• Grant all preceding permissions.</li> </ul>

Item	Description
<b>Authorization for resources, instances, jobs, and volumes</b>	<p>The following permissions can be granted on a resource, instance, job, or volume:</p> <ul style="list-style-type: none"><li>• Read.</li><li>• Update.</li><li>• Delete.</li><li>• Grant all preceding permissions.</li></ul>

Item	Description
Sandbox protection	<p>The sandbox mechanism can restrict access to system resources in MapReduce and UDF programs. Specific restrictions are as follows:</p> <ul style="list-style-type: none"> <li>• Direct access to local files is not allowed. You can only read resource information and generate log information through System.out and System.err.</li> </ul> <div data-bbox="877 784 1434 1030">  <b>Note:</b>  You can view log information by running the Log command on the MaxCompute client. </div> <ul style="list-style-type: none"> <li>• Direct access to Apsara Distributed File System is not allowed.</li> <li>• JNI calls are not allowed.</li> <li>• Java threads cannot be created, and Linux commands cannot be executed by sub-threads.</li> <li>• Network access operations such as acquiring local IP addresses are not allowed.</li> <li>• Java reflection is not allowed. You cannot force access to protected or private members to be valid.</li> </ul>
Control over the quotas of storage and computing resources	<p>Supported. You can limit the number of files and used disk capacity in a project . You can also use quotas to limit the available CPU and memory capacity of the project.</p>

### 23.1.5.1.7 Resource management and task scheduling

Table 23-7: Specifications

Item	Description
File count quota and storage capacity quota	The quotas vary with projects.
Configuration of CPU quota for a resource group	You can configure the minimum or maximum number of virtual CPUs that can be used by a resource group.
Configuration of memory quota for a resource group	You can configure the minimum or maximum amount of virtual memory that can be used by a resource group.
Resource preemption	Preemption of resources within a quota group is supported.
Task scheduling methods	Fair scheduling and first-in-first-out (FIFO).
Configuration of task priorities	By default, task priorities are assigned in a project. You can configure the priorities as needed.
Restart of a failed task	Supported.
Speculative execution of a task	Supported.

### 23.1.5.1.8 Data tables

Table 23-8: Specifications

Item	Description
Data storage methods	CFile data exclusive to MaxCompute is stored in columns in Apsara Distributed File System.
Data compression	Supported. The efficiency of compression is dependent on the data format. The compression ratio between the original and compressed data is 3:1. Infrequently accessed data can be archived in RAID to reduce the storage space it occupies by 50%.
Lifecycle	Supported.
Basic data types	BigInt, String, Boolean, Double, DateTime, and Decimal.

Item	Description
Partitions	Supported. Only String type partitions are supported.
Maximum number of columns	1,024
Maximum number of partitions	60,000
Partition levels	A table can contain up to five partition levels.
Views	Supported. A view can only contain one valid SELECT statement. Materialized views are not supported.
Statistics	Supported. You can define statistical metrics for data tables and view, analyze, and delete statistics.
Comments	Supported. You can make comments for both tables and columns. Comments can be up to 1024 characters in length.

### 23.1.5.1.9 SQL

#### 23.1.5.1.9.1 DDL


Table 23-9: Specifications

Item	Description
Creation of tables	Supported.
Deletion of tables	Supported.
Renaming of tables	Supported.
Creation of views	Supported.
Deletion of views	Supported.
Renaming of views	Supported.
Adding of partitions	Supported.
Deletion of partitions	Supported.
Adding of columns	Supported.
Modification of column names	Supported.
Modification of comments	Supported. You can modify comments for tables and columns.


Item	Description
Modification of the lifecycle of tables	Supported.
Disabling of the lifecycle for specific table partitions	Supported. The command syntax is as follows: <pre>ALTER TABLE table_name [partition_spec] ENABLE DISABLE LIFECYCLE</pre>
Emptying of data from non-partitioned tables	Supported. The command syntax is as follows: <pre>TRUNCATE TABLE table_name</pre>
Modification of table owners	Supported.
Modification of the time when a table or partition was last modified	Supported.

### 23.1.5.1.9.2 DML

Table 23-10: Specifications

Item	Description
Dynamic partition filtering	Supported. This technique can reduce the amount of data to be read. The command syntax is as follows: <pre>select_statement FROM from_statement WHERE PT1 IN (SUBQUERY) AND PT2 IN ( SUBQUERY)... ;</pre>
Multiple outputs	Supported. A single SQL statement can contain up to 128 outputs.  <b>Note:</b> In each output, you can only specify once whether to target a partition in a partitioned table or target a non-partitioned table.
Data update and overwriting	Supported. Batch update is supported.
Aggregation	Supported.



Item	Description
Sorting	Supported. Sorting must be performed with the limit syntax.
Nested subqueries	Supported.
Joins	Supported. SQL joins include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.
UNION ALL	Supported.
CASE WHEN	Supported.
Relational operations	Supported.
Mathematical operations	Supported.
Logical operations	Supported.
Implicit conversions	Supported.
MAPJOIN	<p>Supported. To speed the JOIN operation when volume of data is small, SQL loads all specified small tables into the memory of a program executing the JOIN operation. The default maximum data size is 512 MB. The maximum size cannot exceed 2 GB. Up to six small tables can be specified.</p> <div>  <b>Note:</b>  Take note of the following limits: <ul style="list-style-type: none"> <li>• The left table of a LEFT OUTER JOIN clause must be a large table.</li> <li>• The right table of a RIGHT OUTER JOIN clause must be a large table.</li> <li>• Both the left and right tables of an INNER JOIN clause can be large tables.</li> <li>• MAPJOIN cannot be used in a FULL OUTER JOIN clause.</li> <li>• MAPJOIN supports small tables as subqueries.</li> <li>• When MAPJOIN is used and a small table or subquery is referenced, you must reference the alias of the small table or subquery.</li> <li>• MAPJOIN supports both non-equivalent JOIN conditions and multiple conditions connected by using OR statements.</li> </ul> </div>

Item	Description
Query of the execution plans of DML statements	Supported. The description of the final execution plan corresponding to a DML statement can be displayed. The command syntax is as follows:  <pre>EXPLAIN &lt;DML query&gt;;</pre>

### 23.1.5.1.9.3 Built-in functions

Table 23-11: Specifications

Item	Description
Built-in functions	Supported. Built-in functions include string functions, date functions, mathematical functions, regular functions, and window functions.

### 23.1.5.1.9.4 User-defined functions

Table 23-12: Specifications

Item	Description
Scalar functions	Supported. You can use the Java SDK and Python SDK to write scalar functions.
Aggregate functions	Supported. You can use the Java SDK and Python SDK to write aggregate functions.
Table functions	Supported. You can use the Java SDK and Python SDK to write table functions .
Implicit conversions	Supported.

## 23.1.5.1.10 MapReduce

### 23.1.5.1.10.1 Programming support

Table 23-13: Specifications

Item	Description
Java language	Supported.

Item	Description
Standalone debugging mode	Supported.
Extended MapReduce model	Supported. A Map operation can be followed by any number of Reduce operations. Example: Map-Reduce-Reduce.

### 23.1.5.1.10.2 Job size

Table 23-14: Specifications

Item	Description
Maximum number of mappers	100,000
Maximum number of reducers	2,000
Setting of the number of mappers and reducers	Supported. You can change the number of mappers by changing the input volume of each Map worker. By default, the number of reducers is set at 25% of the number of mappers. You can change this proportion to suit your business needs.
Setting of the memory of mappers and reducers	Supported. The default memory of a mapper or reducer is 2 GB.




**Note:**

The maximum numbers of mappers and reducers are related to the cluster size.

### 23.1.5.1.10.3 Input and output

Table 23-15: Specifications

Item	Description
Input and output of a table	Supported.
Processing of unstructured data	Supported. Volumes are suited to store unstructured data. MaxCompute MapReduce can be used to process unstructured data.
Input and output of multiple tables	Supported. The numbers of inputs and outputs cannot exceed 128.

Item	Description
Reading of resources	<p><b>Supported.</b> A single task can reference up to 256 resources. The total size of all referenced resources cannot exceed 2 GB.</p> <div>  <b>Note:</b>  The maximum number of read attempts for a resource is 64. </div>

#### 23.1.5.1.10.4 MapReduce computing

Table 23-16: Specifications

Item	Description
Custom setup, map, and cleanup methods of mappers	<b>Supported.</b>
Custom setup, reduce, and cleanup methods of reducers	<b>Supported.</b> Transmitted messages are processed in the next iteration.
Custom partition columns or partitions	<b>Supported.</b>
Configuration of mapper output columns to be sorted and grouped by keys	<b>Supported.</b> Note that custom key comparators are not supported.
Custom combiners	<b>Supported.</b>
Custom counters	<b>Supported.</b> A single job cannot have more than 64 custom counters.
Map-only jobs	<b>Supported.</b> To implement Map-only jobs, set the number of Reduce jobs to 0.
Configuration of job priorities	<b>Supported.</b>

#### 23.1.5.1.11 Graph

##### 23.1.5.1.11.1 Programming support

Table 23-17: Specifications

Item	Description
Java language	<b>Supported.</b>

Item	Description
Standalone debugging mode	Supported.

### 23.1.5.1.11.2 Job size

Table 23-18: Specifications

Item	Description
Maximum number of concurrent workers	1,000
Custom worker CPU and memory	Supported. By default, a worker has two CPU cores and 4 GB of memory. A worker can have up to eight CPU cores and 12 GB of memory.

### 23.1.5.1.11.3 Graph loading

Table 23-19: Specifications

Item	Description
Loading of graph data from MaxCompute tables	Supported.
Division of graphs by vertex	Supported.
Custom partitioners	Supported.
Custom split size	Supported. The default split size is 64 MB.
Custom conflict logic upon data loading	Supported. For example, creating duplicate vertices and edges is considered a conflict logic.

### 23.1.5.1.11.4 Iterative computing

Table 23-20: Specifications

Item	Description
Bulk Synchronous Parallel (BSP) computing model	Supported.
Transmission of messages between vertices	Supported. Transmitted messages are processed in the next iteration.

Item	Description
Multiple iteration termination conditions	<ol style="list-style-type: none"> <li>1. The maximum number of iterations is reached.</li> <li>2. All vertices enter the halted state.</li> <li>3. An aggregator determines to terminate the iteration.</li> </ol>
Automatic checkpoint mechanism	Supported.
Custom aggregators	Supported.
Custom combiners	Supported.
Custom counters	Supported. A single job cannot have more than 64 custom counters.
Custom conflict logic	Supported. For example, sending messages to a non-existent vertex is considered a conflict logic.
Writing of computing results to MaxCompute tables	Supported.
Configuration of job priorities	Supported.

### 23.1.5.1.12 Processing of unstructured data

#### *23.1.5.1.12.1 Processing of Table Store data*

Table 23-21: Specifications

Item	Description
Table Store data types	A variety of data types are supported.

#### *23.1.5.1.12.2 Processing of OSS data*

Table 23-22: Specifications

Item	Description
User-defined split and range functions	Supported.
User-defined maximum number of concurrent mappers	Supported.
User-defined file list	Supported.

### 23.1.5.1.12.3 Multiple data sources

Table 23-23: Specifications

Item	Description
Support for various open-source data formats through the STORED AS syntax	Supported data formats include PARQUET, ORC, SEQUENCEFILE, TEXTFILE, and AVRO.

### 23.1.5.1.13 Spark on MaxCompute

#### 23.1.5.1.13.1 Programming support

Table 23-24: Specifications

Item	Description
Native Apache Spark APIs	Supported. You can use native Spark APIs to write code and process data stored in MaxCompute.
Native methods to submit Spark jobs	Supported.
Multiple native Spark components	Spark SQL, Spark MLlib, GraphX, and Spark Streaming are currently supported.
Multiple programming languages	MaxCompute data can be processed using Scala, Python, Java, and R languages.

#### 23.1.5.1.13.2 Data sources

Table 23-25: Specifications

Item	Description
Processing of unstructured data	Supported. You can use Spark APIs to write code and process data stored in OSS and Table Store.
Processing of data from MaxCompute tables and resources	Supported.

### 23.1.5.1.13.3 Scalability

Table 23-26: Specifications

Item	Description
Deep integration of Spark and MaxCompute	Supported. Spark and MaxCompute share cluster resources. Spark resources can be scaled from large-scale MaxCompute clusters.

### 23.1.5.1.14 Elasticsearch on MaxCompute

#### 23.1.5.1.14.1 Programming support

Table 23-27: Specifications

Item	Description
Native Elasticsearch APIs	Supported.

#### 23.1.5.1.14.2 System capabilities

Table 23-28: Specifications

Item	Description
Real-time analysis and retrieval of data at the petabyte level	Supported.
Web-based display for basic server metrics	Supported. A user-friendly O&M platform for index databases and full-text retrieval clusters can be used to monitor the status of index databases and machines in real time.
Data snapshot technology based on Apsara Distributed File System	Supported. Rapid data backup and recovery can be performed to ensure data reliability.
Millisecond-level response to keyword-based and comprehensive searches and second-level response to fuzzy searches	Supported.
Real-time analysis and retrieval of imported data and query response times within 500 milliseconds	Supported. The storage architecture is powered by the distributed cache-accelerated block device technology.



Item	Description
In-memory off-heap storage and processing of index data and fine-grained memory management	Supported.

### 23.1.5.1.15 Other extensions

The following extended plug-ins and tools are both client-specific and open-source. You can download the plug-ins and tools at <https://github.com/aliyun/>.

Table 23-29: Specifications

Item	Description
R language support	RODPS is a plug-in for the MaxCompute client to support the R language.
Plug-ins and tools	Eclipse plug-ins and command line tools are available .
OGG	OGG plug-ins synchronize data from OGG to DataHub.
Flume	Flume plug-ins synchronize data from Flume to DataHub.
FluentD	FluentD plug-ins synchronize data from FluentD to DataHub.
JDBC	JDBC interfaces are partially supported.
Sqoop	Sqoop can be used to exchange data with MaxCompute.

### 23.1.5.2 Hardware specifications

The following table lists the hardware specifications of MaxCompute.

Table 23-30: Hardware specifications

Node type	Server configuration	Number of nodes	Description
Management node	<ul style="list-style-type: none"><li>• CPU: dual-socket 8-core or higher</li><li>• Memory: 256 GB or higher</li><li>• Disk: two 4 TB NVMe U.2 SSDs</li><li>• NIC: two 10 GE NICs for network bonding</li></ul>	/	We recommend that you use Intel Platinum 81xx series processors and higher configurations. <ul style="list-style-type: none"><li>•</li></ul>

Node type	Server configuration	Number of nodes	Description
<b>Control node</b>	<ul style="list-style-type: none"> <li>• <b>CPU:</b> dual-socket 8-core or higher</li> <li>• <b>Memory:</b> 128 GB or higher</li> <li>• <b>Disk:</b> one 4 TB SATA HDD with 7200 RPM performance</li> <li>• <b>NIC:</b> two 10 GE NICs for network bonding</li> </ul>	8/13	<ul style="list-style-type: none"> <li>• We recommend that you use Intel Platinum 81xx series processors and higher configurations.</li> <li>• When the number of data nodes is less than 500, the number of control nodes is 8. When the number of data nodes is more than 500, the number of control nodes is 13.</li> <li>• We recommend that you deploy data nodes in containers when the number of data nodes is less than 500.</li> <li>• When all control nodes are physical servers and the number of data nodes is less than 1,000, you can implement a hybrid deployment of control nodes and data nodes based on your actual needs.</li> <li>• The system disk capacity is greater than or equal to 240 GB.</li> </ul>

Node type	Server configuration	Number of nodes	Description
Hybrid deployment of management nodes and control nodes	<ul style="list-style-type: none"> <li>• CPU: dual-socket 8-core or higher</li> <li>• Memory: 256 GB or higher</li> <li>• Disk: one 4 TB NVMe U.2 SSDs</li> <li>• NIC: two 10 GE NICs for network bonding</li> </ul>	/	<ul style="list-style-type: none"> <li>• Hybrid deployment is recommended when the number of data nodes is less than 500 and is not expected to be increased.</li> <li>• Assume that the number of data nodes is approximately 500 and is expected to be increased to more than 500. When you deploy the nodes for the first time, we recommend that you deploy them separately on physical servers.</li> </ul>

Node type	Server configuration	Number of nodes	Description
Data node	<ul style="list-style-type: none"> <li>• CPU: dual-socket 8-core or higher</li> <li>• Memory: 128 GB or higher</li> <li>• Disk: twelve 2 TB, 4 TB, 6 TB, or 8 TB SATA HDDs with 7200 RPM performance</li> <li>• NIC: two 10 GE NICs for network bonding</li> </ul>	Depends on the amount of data.	<ul style="list-style-type: none"> <li>• We recommend that you use Intel Golden 61xx series processors and higher configurations.</li> <li>• The recommended ratio of core quantity to memory capacity is 1:4.</li> <li>• We recommend that you add a 4 TB NVMe U.2 SSD when the number of cores is greater than or equal to 48.</li> <li>• Number of nodes = <math>\lceil \frac{(\text{Total planned data volume} \times \text{Data expanding rate (1.3)} \times \text{Data compression rate (1)} \times \text{Number of replicas (3)})}{\text{Disk utilization rate (0.85)} / \text{Disk formatting loss (0.9)} / ((\text{Number of disks (12)} - \text{Number of system reserved blocks (1)}) \times \text{Disk capacity (8 TB)})} \rceil</math> rounded up.</li> </ul>

**Note:**

- We recommend that you use the preceding configurations in offline scenarios as needed.

- We recommend that you do not use two or more machine types for compute nodes of MaxCompute.
- We recommend that you do not use both 1 GE and 10 GE NICs for MaxCompute.
- The configuration of machines to be added cannot be lower than that of the existing machines.
- The utilization of used compute nodes needs to be evaluated together with the business side.

### 23.1.5.3 Specifications of DNS resources

Table 23-31: Specifications

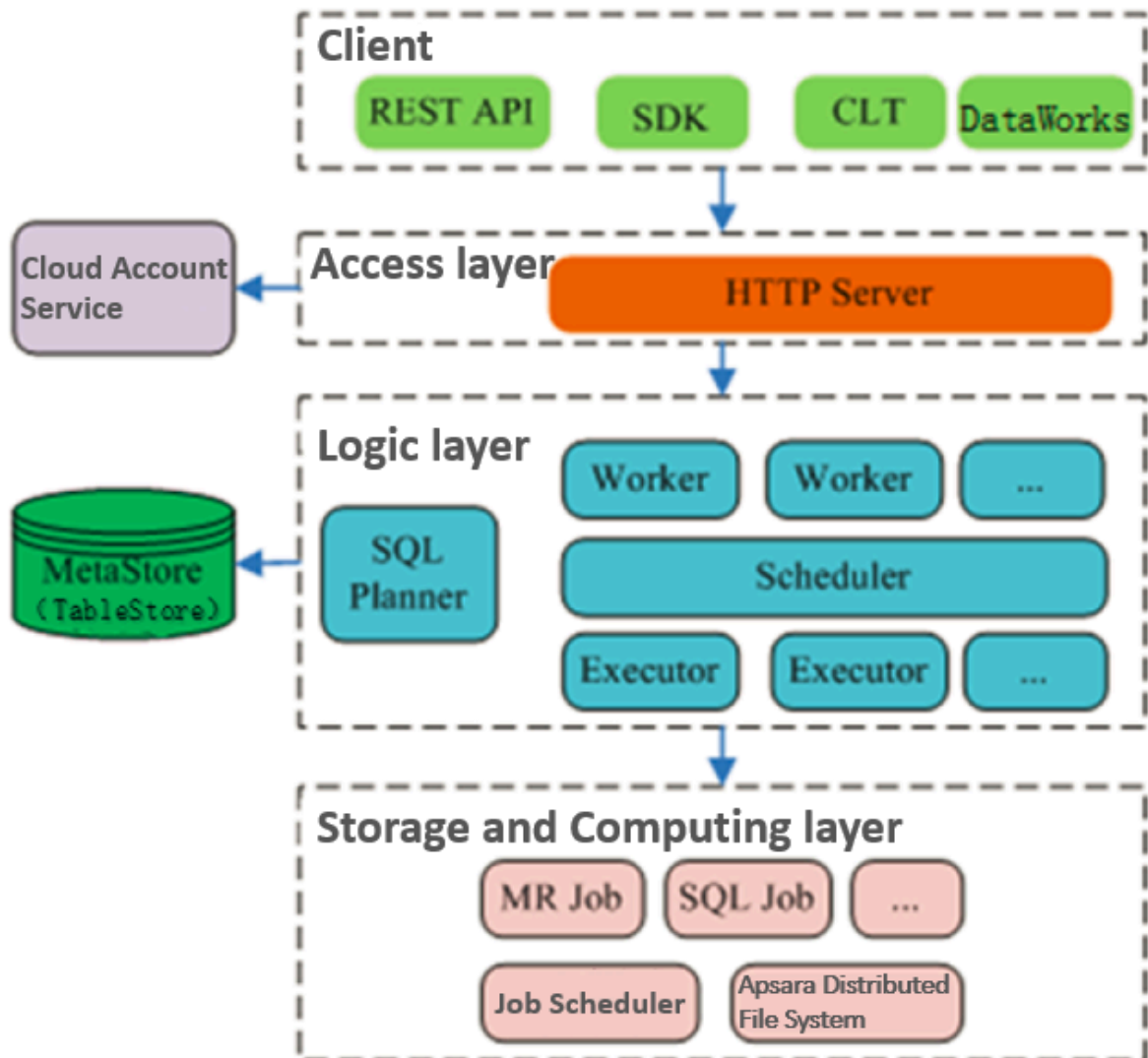
Resource name	Domain name	Description
odps_frontend	odps_frontend_server_inner_dns	The internal domain name of the MaxCompute front-end server. This domain name is not subject to VPC.
	odps_frontend_server_public_dns	The private domain name of the MaxCompute front-end server.
	odps_frontend_server_internet_dns	The public domain name of the MaxCompute front-end server.
tunnel_frontend	odps_tunnel_frontend_server_inner_vip	The internal domain name of the front-end server for MaxCompute Tunnel. This domain name is not subject to VPC.
	odps_tunnel_frontend_server_public_vip	The private domain name of the front-end server for MaxCompute Tunnel.

Resource name	Domain name	Description
	<b>odps_tunnel_frontend_server_in ternet_vip</b>	<b>The public domain name of the front -end server for MaxCompute Tunnel.</b>
<b>cupid_web_proxy</b>	<b>odps_jobview_dns</b>	<b>The internal domain name of the MaxCompute Jobview . This domain name is not subject to VPC.</b>
<b>logview</b>	<b>odps_logview_inner_dns</b>	<b>The internal domain name of the MaxCompute Logview . This domain name is not subject to VPC.</b>
	<b>odps_logview_public_dns</b>	<b>The private domain name of the MaxCompute Logview.</b>
<b>web_console</b>	<b>odps_webconsole_inner_dns</b>	<b>The internal domain name of the MaxCompute Web console. This domain name is not subject to VPC.</b>
	<b>odps_webconsole_public_dns</b>	<b>The private domain name of the MaxCompute Web console.</b>

## 23.2 Architecture

*Figure 23-5: MaxCompute architecture* shows the MaxCompute architecture.

Figure 23-5: MaxCompute architecture



The MaxCompute service is divided into four parts: client, access layer, logic layer, and storage and computing layer. Each layer can be scaled out.

The following methods can be used to implement the functions of a MaxCompute client:

- **API:** RESTful APIs are used to provide offline data processing services.
- **SDK:** RESTful APIs are encapsulated in SDKs. SDKs are currently available in programming languages such as Java.



- **Command line tool (CLT):** This client-side tool runs on Windows and Linux. CLT allows you to submit commands to manage projects and use DDL and DML.
- **DataWorks:** DataWorks provides upper-layer visual ETL and BI tools that allow you to synchronize data, schedule tasks, and create reports.

The access layer of MaxCompute supports HTTP, HTTPS, load balancing, user authentication, and service-level access control.

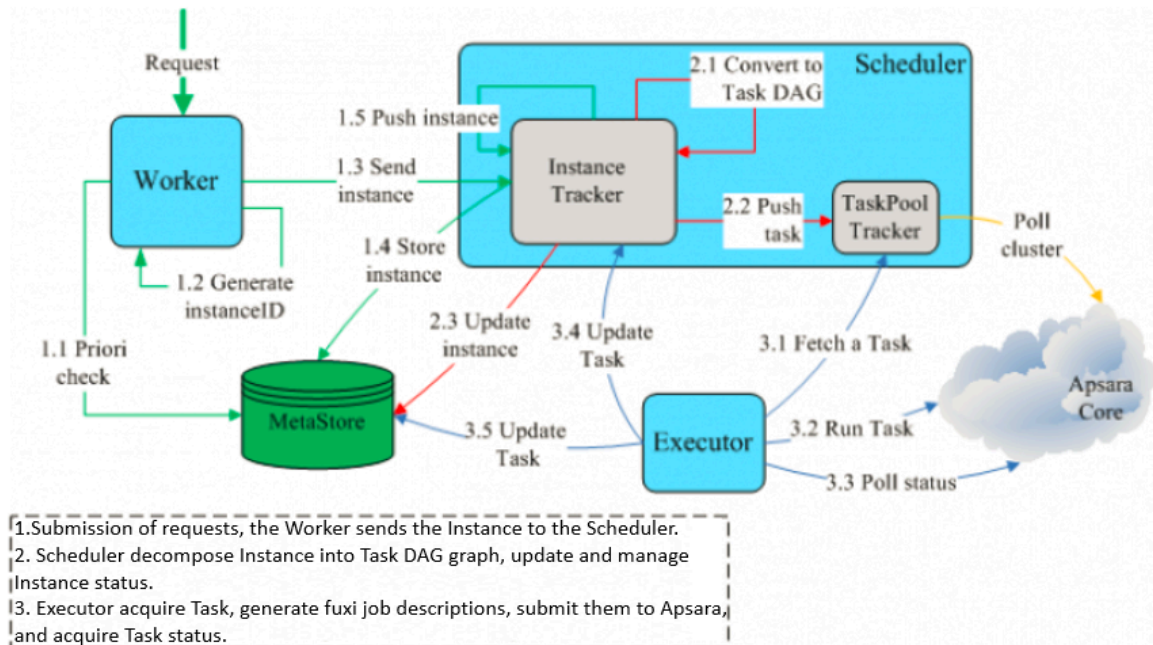
The logic layer is at the core of MaxCompute. It supports project and object management, command parsing and execution logic, and data object access control and authorization. The logic layer is divided into control and compute clusters. The control cluster manages projects and objects, parses queries and commands, and authorizes access to data objects. The compute cluster executes tasks. Both control and compute clusters can be scaled out as required. The control cluster is comprised of three different roles: Worker, Scheduler, and Executor. These roles are described as follows:

- The Worker role processes all RESTful requests and manages projects, resources, and jobs. Workers forward jobs that need to launch Fuxi tasks (such as SQL, MapReduce, and Graph jobs) to the Scheduler for further processing.
- The Scheduler role schedules instances, splits instances into multiple tasks, sorts tasks that are pending for submission, and queries resource usage from FuxiMaster in the compute cluster for throttling. If there are no idle slots in Job Scheduler, the Scheduler stops processing task requests from Executors.
- The Executor role is responsible for launching SQL and MapReduce tasks. Executors submit Fuxi tasks to FuxiMaster in the compute cluster and monitor the operating status of these tasks.

In summary, when you submit a job request, the Web server at the access layer queries the IP addresses of registered Workers and sends API requests to randomly selected Workers. The Workers then send these requests to the Scheduler for scheduling and throttling. Executors actively poll the Scheduler queue. If the necessary resources are available, the Executors start executing tasks and

return the task execution status to the Scheduler. The following figure shows the corresponding business execution logic.

Figure 23-6: Business execution logic



The storage and computing layer of MaxCompute is a core component of the proprietary cloud computing platform developed by Alibaba Cloud. As the kernel of the Apsara system, this layer runs in the compute cluster independent of the control cluster. The preceding MaxCompute architecture diagram illustrates only major modules of the Apsara kernel, such as Apsara Distributed File System and Job Scheduler.

Among the modules, Apsara Distributed File System is designed to aggregate the storage resources of a large number of machines and provide users with reliable large-scale distributed storage services. Apsara Distributed File System is an important part of the Apsara kernel.

Apsara Distributed File System includes three masters and multiple chunkservers. A master is responsible for the storage and management of file metadata, while a chunkserver is responsible for data storage. Identical blocks of data are stored on multiple chunkservers to ensure its reliability. In normal cases, data is stored in Apsara Distributed File System in three copies. All MaxCompute data files are stored in Apsara Distributed File System. The data files can be found in the /

product/aliyun/odps directory. It is important to note that masters operate in hot standby mode. Only one master operates at a time.

- **master:**
  1. PanguMaster maintains metadata for the entire file system, including namespaces, file-to-block mappings, and data block storage addresses.
  2. PanguMaster is the heart of the distributed file storage system and controls system-level activities such as garbage collection for isolated data blocks, data merging among chunkservers, chunkserver health check, and recovery of data blocks lost due to a down chunkserver.
  3. PanguMaster also manages data access requests originating from multiple clients at the same time to ensure the integrity of data in the cluster.
  4. PanguMaster only allows the client to perform operations on metadata. Data transmissions are conducted directly among chunkservers.
- **chunkserver:** The files in Apsara Distributed File System are divided into fixed-size storage units called chunks. A machine that stores chunks is called a chunkserver. PanguMaster assigns a 128-bit ID to a chunk when the chunk is created. The Apsara Distributed File System client reads the chunks stored in disks based on chunk IDs.

To provide better support for the processing of structured data in MaxCompute tables, the MaxCompute team has implemented a special Apsara Distributed File System file format called CFile.

- CFile is a file format based on column storage. It is designed to reduce invalid disk read operations during offline data processing. Data in the file is clustered by column into blocks and compressed to reduce storage space. This means that in offline data processing scenarios, you only need to read the required data. This avoids unnecessary disk operations, improves disk read efficiency, and reduces network bandwidth consumption.
- The CFile storage structure can be logically divided into three areas: data area, index area, and header area. The data area stores the user data that is clustered by column and uses blocks as organizational units. The index area stores the indexes corresponding to the data blocks of each column, which includes the starting position of each block in the file, the length of a compressed block, and the data amount in a block (for variable-length data types such as string ). The header area stores the metadata of each column in the file, such as the

starting position of the column index, index length, column type information, compression method, and row count and version of user data.

- **MaxCompute supports the following data types:**
  - **Bigint:** represents an 8-byte signed integer.
  - **Boolean:** represents a logical true or false value.
  - **Double:** represents an 8-byte double-precision floating-point number.
  - **String:** represents a string in UTF-8 format. MaxCompute functions automatically assume that string objects contain UTF-8 encoded strings. If the string is encoded in other formats, an error occurs.
  - **Datetime:** represents a date and time in the YYYY-MM-DD HH:mm:ss format.  
Example: 2012-01-02 10:09:25

**Job Scheduler** is a module for resource management and task scheduling in the Apsara kernel. It also provides a basic programming framework for application development. It is designed to make full use of the hardware resources of the entire cluster to meet the computing requirements of users and systems. Job Scheduler supports the processing of two application types: a low-latency online service called FuxiService and a high-throughput offline processing application called FuxiJob. Job Scheduler is similar to YARN in Hadoop.

- **FuxiService:** a resident process in Job Scheduler. You can send requests to create and destroy a service. Job Scheduler does not proactively destroy a service process.
- **FuxiJob:** a temporary task in Job Scheduler. When a task ends, the resources are released and reclaimed by Job Scheduler.

**Job Scheduler** schedules and allocates cluster storage and computing resources to upper-layer applications. Job Scheduler is able to manage computing resource quotas, access control policies, and job priorities to ensure that resources are shared effectively. Job Scheduler provides a data-driven, multi-level parallel computing framework, which is similar to the MapReduce programming model. The framework is ideal for complex applications such as large-scale data processing and large-scale computing.

**Job Scheduler** has two masters and multiple Tubos. Masters operate in cold standby mode. Only one master operates at a time. A Turbo process is started on each compute node to manage available resources of each machine, such as the CPU,

memory, hard disk, and network, and record the resources used on each machine. The Turbo process on each machine reports the resource usage to FuxiMaster, which centrally manages and schedules the resources.

## 23.3 Features

### 23.3.1 Tunnel

#### 23.3.1.1 Overview

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.

The following table lists the major APIs.

Table 23-32: Major APIs

API	Description
<b>TableTunnel</b>	The entry class of the MaxCompute Tunnel service.
<b>TableTunnel.UploadSession</b>	The session that uploads data to a MaxCompute table.
<b>TableTunnel.DownloadSession</b>	The session that downloads data from a MaxCompute table.



**Notice:**

The tunnel endpoint supports automatic routing based on the MaxCompute endpoint settings.

#### 23.3.1.2 TableTunnel

This topic describes the TableTunnel API.

**API definition:**

```
public class TableTunnel {
    public DownloadSession createDownloadSession(String projectName,
        String tableName);
    public DownloadSession createDownloadSession(String projectName,
        String tableName, PartitionSpec partitionSpec);
}
```

```

public UploadSession createUploadSession(String projectName, String
tableName);
public UploadSession createUploadSession(String projectName, String
tableName, PartitionSpec partitionSpec);
public DownloadSession getDownloadSession(String projectName, String
tableName, PartitionSpec partitionSpec, String id);
public DownloadSession getDownloadSession(String projectName, String
tableName, String id);
public UploadSession getUploadSession(String projectName, String
tableName, PartitionSpec partitionSpec, String id);
public UploadSession getUploadSession(String projectName, String
tableName, String id); public void setEndpoint(String endpoint);
}

```

#### TableTunnel API description:

- **Lifecycle:** From the creation of the TableTunnel instance to the end of the program.
- **Purpose:** It provides a method to create Upload and Download objects.

### 23.3.1.3 UploadSession

This topic describes the UploadSession API.

#### API definition:

```

public class UploadSession {
UploadSession(Configuration conf, String projectName, String tableName
, String partitionSpec) throws TunnelException;
UploadSession(Configuration conf, String projectName, String tableName
, String partitionSpec, String uploadId) throws TunnelException;
public void commit(Long[] blocks); public Long[] getBlockList();
public String getId();
public TableSchema getSchema();
public UploadSession.Status getStatus(); public Record newRecord();
public RecordWriter openRecordWriter(long blockId);
public RecordWriter openRecordWriter(long blockId, boolean compress);
}

```

#### UploadSession API description.

Table 23-33: UploadSession API

Item	Description
<b>Lifecycle</b>	<b>From the upload instance creation to the end of the uploading.</b>
<b>Purpose</b>	<p><b>Creates an upload instance by calling a constructor method or by using the TableTunnel class.</b></p> <ul style="list-style-type: none"> <li>• <b>Request mode: synchronous.</b></li> <li>• <b>The server creates an upload session and generates a unique upload ID. You can get the upload ID by running getId on the console.</b></li> </ul>

Item	Description
Upload data	<ul style="list-style-type: none"> <li>• Request mode: asynchronous.</li> <li>• Call <code>openRecordWriter</code> to generate a <code>RecordWriter</code> instance. The <code>blockId</code> parameter identifies the data to upload this time and the position of the data in the table. The value range is [0, 20000]. In case the uploading fails, the data is re-uploaded based on the block ID.</li> </ul>
Check uploading	<ul style="list-style-type: none"> <li>• Request mode: synchronous.</li> <li>• Call <code>getStatus</code> to get the uploading status.</li> <li>• Call <code>getBlockList</code> to get a list of the block IDs of successful uploading instances, check the block ID list, and re-upload data for failed uploading instances.</li> </ul>
Stop uploading	<ul style="list-style-type: none"> <li>• Request mode: synchronous.</li> <li>• Call <code>commit(Long[] blocks)</code>. The <code>blocks</code> parameter indicates the list of block IDs of successful uploading instances. The server will verify the block ID list.</li> <li>• The verification improves data correctness. If the provided block list is different from the block list on the server, an error is reported.</li> </ul>
Status	<ul style="list-style-type: none"> <li>• UNKNOWN: Initial value set while server just creates a session.</li> <li>• NORMAL: An UPLOAD object is created successfully.</li> <li>• CLOSING: The server sets the upload session to CLOSING status before calling the COMPLETE method (to complete uploading).</li> <li>• CLOSED: The uploading is completed (data has been moved to the directory where the result table is).</li> <li>• EXPIRED: The upload session is timed out.</li> <li>• CRITICAL: An error occurs.</li> </ul>

**Notice:**

- `blockId` in the same `UploadSession` API must be unique. That is, after a block ID is used to start `RecordWriter` in an upload session, data is written, and the session is closed and committed, this block ID cannot be used to start another `RecordWriter`.

- The maximum size of a block is 100 GB. We strongly recommend that 64 MB or more data is written into each block. Otherwise, the computing performance will seriously degrade.
- Each session has a 24-hour life cycle on the server.
- You are advised to have data prepared before calling `openRecordWriter`. A network action is triggered every time the Writer writes 8 KB data. If no network action is triggered in the last 120 seconds, the server closes the connection and the Writer becomes unavailable. You have to start a new Writer.

### 23.3.1.4 DownloadSession

This topic describes the `DownloadSession` class.

**API definition:**

```
public class DownloadSession {
    DownloadSession(Configuration conf, String projectName, String
tableName, String partitionSpec) throws TunnelException
    DownloadSession(Configuration conf, String projectName, String
tableName, String partitionSpec, String downloadId) throws TunnelExce
ption
    public String getId()
    public long getRecordCount() public TableSchema getSchema()
    public DownloadSession.Status getStatus()
    public RecordReader openRecordReader(long start, long count)
    public RecordReader openRecordReader(long start, long count, boolean
compress)
}
```

**DownloadSession API description.**

Table 23-34: DownloadSession API

Parameter	Description
<b>Lifecycle</b>	From the creation of the Download instance to the end of the download process.
<b>Purpose</b>	<p>Creates a Download instance by calling a constructor method or using <code>TableTunnel</code>.</p> <ul style="list-style-type: none"> <li>• Request mode: Synchronous.</li> <li>• The server creates a session for this Download and generates a unique download ID to mark the Download. The console can get data with a get ID. The operation has a high overhead. The server creates indexes for the data files. If many data files exist, the operation takes a long time. Then the server returns the total number of records, and starts concurrent downloads according to the number of records.</li> </ul>



Parameter	Description
Download data	<ul style="list-style-type: none"> <li>• <b>Request mode: Asynchronous.</b></li> <li>• <b>Call openRecordReader to generate a RecordReader instance. The Start parameter marks the start position of record for this download. The value of Start is equivalent to or greater than 0. The Count parameter marks the number of records for this download. The value of Count is greater than 0.</b></li> </ul>
View the download process	<ul style="list-style-type: none"> <li>• <b>Request mode: Synchronous.</b></li> <li>• <b>Call getStatus to get the download status.</b></li> </ul>
Status	<ul style="list-style-type: none"> <li>• <b>UNKNOWN: the initial value that is set when the server creates a session.</b></li> <li>• <b>NORMAL: The download object is successfully created.</b></li> <li>• <b>CLOSED: The download session is completed.</b></li> <li>• <b>EXPIRED: The download session times out.</b></li> </ul>

### 23.3.2 SQL

MaxCompute SQL is a structured query language whose syntax is similar to Oracle, MySQL, and Hive SQL. MaxCompute SQL can be regarded 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.

MaxCompute SQL is applicable to scenarios that have large amounts 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.

### 23.3.3 MapReduce

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).

You can use Java APIs, which are provided by MapReduce, to write MapReduce programs for processing data in MaxCompute. The Map and Reduce concepts are borrowed from functional programming and vector programming languages. This

helps programmers run their programs on distributed systems without performing distributed parallel programming.

MapReduce works only after you specify a Map function and a concurrent Reduce function. The Map function maps a group of key-value pairs to another group of key-value pairs. The Reduce function ensures that all elements in the mapped key-value pairs share the same key group.

MaxCompute MapReduce has the following features:

- It is a Hadoop-style MapReduce function designed for MaxCompute (used to process tables and volumes).
- 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 can read resources.
- It does not allow you to use views as data inputs.
- It provides a limited sandbox security environment.

The following is a detailed procedure of how MapReduce processes data:

1. Before you formally start Map, ensure that `partition` is set for input data. The input data is divided into equal-sized blocks, which are partitions. Each partition is processed as the input of a single Map worker so that multiple Map workers can work together.
2. After partitioning, multiple Map Workers start working simultaneously. Each Map Worker reads its respective shard, computes the shard, and works out the result to Reduce.



**Note:**

During data output, each Map worker needs to specify one key for each output data. The key decides the Reduce worker for which the data is targeted. Multiple keys may correspond to a single Reduce worker. Data of the same key is sent to the same Reduce worker, and a single Reduce worker may receive data with different keys.

3. Before entering the Reduce stage, the MapReduce framework will sort the data Key values to make data with the same Key values adjacent. If you specify

Combiner, the framework will call Combiner and aggregate data with the same Key.



**Note:**

You can customize the Combiner logic. Unlike the typical MapReduce framework protocol, Unlike the typical MapReduce framework protocol, MaxCompute requires the input and output parameters be consistent with those of Reduce. This process is generally called Shuffle.

4. When entering the Reduce stage, data with same Key will be in the same Reduce Worker. A single Reduce Worker may receive data from multiple Map Workers. Each Reduce worker performs the Reduce operation on multiple values with the same key. Finally, the multiple data entries with 1 Key will become 1 Value after the Reduce operation.



**Note:**

The preceding section is only a brief introduction to MapReduce. For more details, see related documentation.

### 23.3.4 Graph

Graph is the computing framework of MaxCompute designed for iterative graph processing. It provides programming interfaces similar to Pregel, allowing you to develop efficient machine learning and data mining algorithms.

Large amounts of data on the Internet is structured as graphs, such as social networking and logistics information. Graph computing models are iterative computing models. Throughout the entire computing process, multiple iterations are performed to achieve convergence. For example, for machine learning algorithms that require iterative learning model parameters, Graph is more suited than MapReduce. In common usage scenarios, you can abstract a question as a graph. Then, you can set the vertex as the center of the graph, and use supersteps for iterative updating.

MaxCompute Graph currently works in two modes:

- **Offline mode:** suitable for large-scale computing. Similar to MapReduce jobs, this mode involves loading and computing.

- **Interactive mode:** suitable for small-scale computing. You can implement a UDF and then use the command line for interaction.

In offline mode, loading and computing are independent processes. Loaded data resides in the memory. You can apply different computing logics to the loaded data. For example, the risk control department may load a set of data once a day. The operations personnel will apply different query logics to the data to view the relationships between the data.

MaxCompute Graph has been applied to many businesses in Alibaba. For example, weighted PageRank algorithms are used to compute influence metrics for Alipay users, and variational Bayesian EM models are used to predict users' car brands based on the properties of the items purchased by users.

### 23.3.5 Unstructured data processing (integrated computing scenarios)

Alibaba Cloud introduced the MaxCompute-based unstructured data processing framework so that MaxCompute SQL commands can directly process external user data, such as unstructured data from OSS. You are no longer required to first import data into MaxCompute tables.

You can run a simple DDL statement to create an external table in MaxCompute, and associate MaxCompute tables with external data sources. This table can then act as an interface between MaxCompute and external data sources. The external table can be accessed in the same way as a MaxCompute table, and computed by MaxCompute SQL.

MaxCompute allows you to process the following data sources by creating external tables:

- **Internal data sources:** OSS, Table Store, AnalyticDB, ApsaraDB for RDS, HDFS (Alibaba Cloud), and TDDL.
- **External data sources:** HDFS (open source), ApsaraDB for MongoDB, and Hbase.

### 23.3.6 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.

## 23.3.7 Enhanced features

### 23.3.7.1 Spark on MaxCompute

#### 23.3.7.1.1 Open-source platform - Cupid

##### *23.3.7.1.1.1 Overview*

MaxCompute is a big data solution independently developed by Alibaba Cloud that leads the industry in scale and stability. The big data open-source communities are actively developing big data solutions. All kinds of systems are rapidly emerging and growing to meet various requirements. To better serve users and to diversify the MaxCompute ecosystem, the MaxCompute team has developed the Cupid platform to connect MaxCompute with open-source communities. The Cupid platform integrates the diversity of open-source communities with the scale and stability of the Apsara system.

The software stacks of open-source communities and the Apsara system are similar with slight differences.

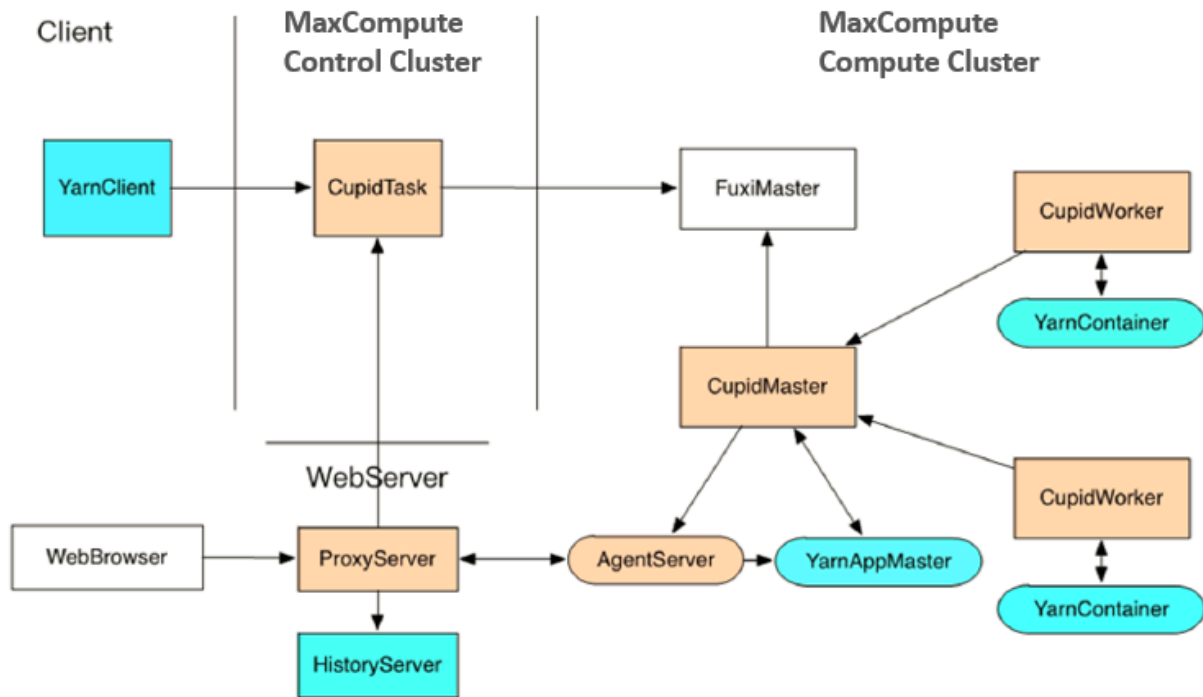
Most open-source communities use HDFS as a distributed file system, while the Apsara system uses Apsara Distributed File System. Most open-source communities use YARN as a distributed scheduling system, while the Apsara system uses Job Scheduler. On top of Job Scheduler are the computing engines designed for all kinds of scenarios. Cupid provides compatibility with open-source communities for open-source applications (such as Spark) to run on MaxCompute.

##### *23.3.7.1.1.2 Compatibility with YARN*

YARN has three application-oriented APIs: YarnClient, AMRMClient, and NMClient. YarnClient is used to submit applications to a cluster. AMRMClient is used by

AppMaster to send messages to Resource Manager to request and release resources. NMClient is used to start and stop application containers.

Figure 23-7: YARN on MaxCompute



The preceding figure shows the process of submitting a YARN application to be run on MaxCompute. The yellow boxes indicate Cupid components, while the light blue boxes indicate open-source components. The procedure is as follows:

1. The transformed and encapsulated YarnClient is used to access the MaxCompute control cluster to submit a job to FuxiMaster by using the transformed Spark client tool.
2. FuxiMaster starts a CupidMaster. Then, the CupidMaster starts YarnAppMaster based on the YARN protocol.
3. YarnAppMaster interacts with FuxiMaster through CupidMaster to request and release resources.
4. To start a new container, you need to use Turbo in Job Scheduler to start a CupidWorker first. The CupidWorker will then start the container based on the YARN protocol.



**Note:**

Typically, YarnAppMaster provides a UI. The UI is implemented through Cupid based on a proxy mechanism.

### ***23.3.7.1.1.3 Compatibility with FileSystem***

Most open-source communities use HDFS as a distributed storage solution. The FileSystem API provided by Hadoop is compatible with Alibaba Cloud OSS and Amazon S3. Apsara Distributed File System is compatible with FileSystem API. Open-source jobs submitted to MaxCompute can be run natively on Apsara Distributed File System.



#### **Note:**

Apsara Distributed File System does not directly provide external services. The data in Apsara Distributed File System can only be used as intermediate job data, such as checkpoints. You can use OSS to make the data stored in Apsara Distributed File System accessible to other environments.

### ***23.3.7.1.1.4 DiskDrive***

Most open-source applications use local file systems for data processing, such as the shuffle and storage modules in Spark. In environments with large clusters, disks are important system resources. Disks must be centrally managed to ensure high availability, performance, and security. In the Apsara system, disks are centrally managed by Apsara Distributed File System. To provide local file system APIs based on Apsara Distributed File System, the Cupid team has designed and implemented the DiskDriverService system by integrating Web-based storage into MaxCompute.

## **23.3.7.1.2 Feature extensions**

### ***23.3.7.1.2.1 Overview***

MaxCompute provides the Cupid framework to support open-source applications. This allows Spark to be run on MaxCompute. For ease of use and better integration with MaxCompute, there are several extensions available for Spark on MaxCompute to add features such as the secure isolation of open-source Spark applications, mutual access between MaxCompute data and Spark data, and support for interactions in multi-tenant clusters.

The following sections describe these extensions.

### ***23.3.7.1.2.2 Security isolation***

Spark jobs submitted to the MaxCompute computing cluster are run in sandboxes , preventing attacks on the system. A parent-child process architecture is used for the entire system. The Cupid framework runs in the parent process, and Spark runs in the child processes. When Spark requires access to system services, the parent process accesses the services on behalf of Spark by communicating with the child processes.

### ***23.3.7.1.2.3 Data interconnection***

An advantage of running Spark on MaxCompute is that the resources used by Spark and MaxCompute jobs are shared across all clusters. This allows jobs to directly access their data without having to pull data across different clusters.

This data includes metadata and storage data. For security reasons, Spark must be authenticated through the MaxCompute account system before it can store MaxCompute data. Spark on MaxCompute provides `OdpsRDD` and `OdpsDataFrame` so that users can use Spark APIs on MaxCompute. Spark SQL has direct access to MaxCompute metadata for SQL optimization and can directly store and retrieve MaxCompute data at the physical layer.

### ***23.3.7.1.2.4 Client mode***

The `yarn-cluster` and `yarn-client` modes are commonly used in open-source communities for Spark-related development efforts. In `yarn-cluster` mode, you can submit a Spark job to a YARN cluster. After the job is run, the client generates a log that indicates the job status. In this mode, you cannot submit a job to a Spark session multiple times in real time, and the client cannot obtain the running status and result of each job. The `yarn-client` mode takes effect for interactive scenarios. However, to use the `yarn-client` mode, you need to launch the Spark driver process from the client side. You cannot use a Spark session as a service in this mode. The MaxCompute team has developed the `Client` mode based on Spark on MaxCompute to solve the preceding problems. The `Client` mode has the following features:

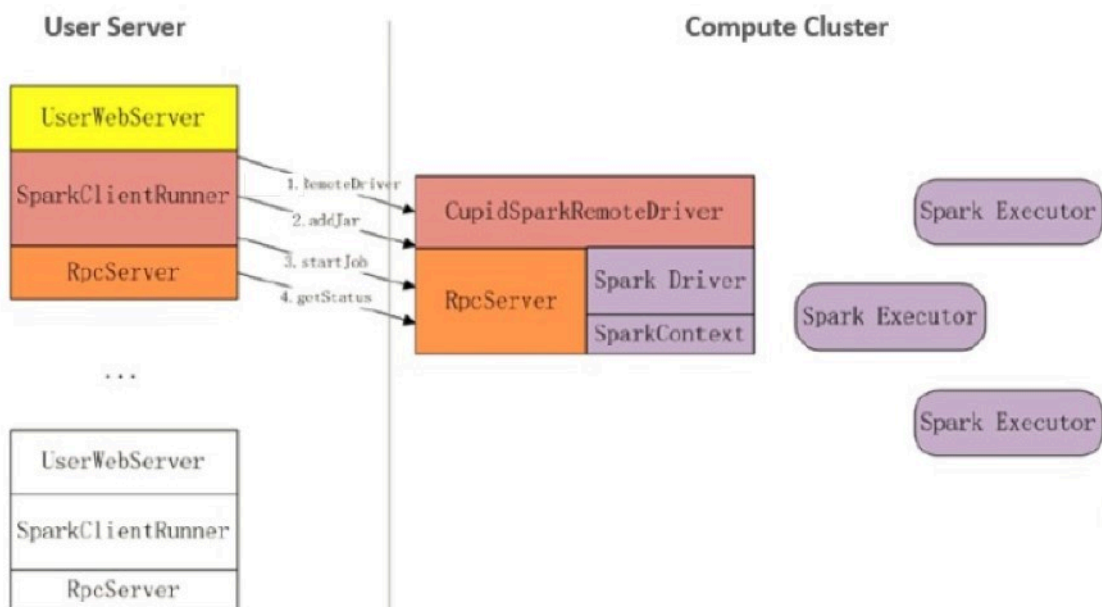
1. The client is a lightweight process that does not require you to launch the Spark driver process.



2. The client provides a set of APIs that can be used to submit jobs in real time to the same Spark session in MaxCompute clusters. It can also monitor the statuses of all jobs in the Spark session.
3. The client can build dependencies between jobs by monitoring job statuses and results.
4. You can compile an application JAR package in real time and submit it to the original Spark session through the client.
5. The client can be integrated into the Web servers of a service, and can also be scaled horizontally.

In **Client mode**, you need to use **CupidSparkClientRunner** to start a Spark session in a MaxCompute cluster. Then, you can use **CupidSparkClientRunner** to perform operations on the client side, such as submitting jobs and viewing the running statuses and results of the jobs. Cached data can be shared between jobs. You can also construct multiple **CupidSparkClientRunner** objects to interact with the same Spark session. The following figure shows the block diagram of the Spark Client mode.

Figure 23-8: Spark Client mode



The procedure for using the Spark Client mode is as follows:

1. You submit a job to a MaxCompute cluster to launch **CupidSparkRemoteDriver** and obtain the **SparkClientRunner** object.

2. You use `SparkClientRunner` to add the JAR package that will execute the job to `RemoteDrive`.
3. `SparkClientRunner` uses the job classname to submit the job to `RemoteDriver`. `RemoteDriver` then runs the job.
4. `SparkClientRunner` monitors the job status based on the job ID returned after the job is submitted.

#### ***23.3.7.1.2.5 Spark ecosystem support***

The Spark ecosystem covers diverse components, including MLlib, Streaming, PySpark, SparkR, GraphX, and SQL. Spark on MaxCompute provides a complete Spark ecosystem that supports the scaling of original resources in open-source communities. The ecosystem provides consistent user experience with that of open-source communities. Spark on MaxCompute also supports access to the Spark UI and historical log files.

### **23.3.7.2 Elasticsearch on MaxCompute**

#### **23.3.7.2.1 Terms**

term

**An exact value that can be indexed. You can use a term query to search for an exact match.**

text

**A piece of unstructured data. Typically, a text is parsed into individual terms that are stored in an Elasticsearch index library.**

cluster

**A collection of one or more nodes that provide external indexing and search services. Elasticsearch is deployed in the Apsara cluster of MaxCompute. Elasticsearch clusters are a part of the Apsara cluster.**

node

**A logical service in an Elasticsearch cluster. A node can store data and participate in the cluster's indexing and search capabilities.**

## shard

**A single Lucene instance which is a relatively low-level feature managed by Elasticsearch. An Elasticsearch cluster automatically manages all the shards in a cluster. When a node fails, Elasticsearch moves the shards to a different node or adds a new node.**

## replica

**A distinct copy in Elasticsearch. Elasticsearch on MaxCompute allows you to have multiple replicas across different nodes to improve system-level availability. We recommend that you set the default number of replicas for this service to 1.**

## index

**A collection of documents that have similar characteristics. For example, you can have an index for customer data, an index for a product catalog, and another index for order data. An index is identified by a name (that must be all lowercase) that is used to refer to the index when you perform indexing, search, update, and delete operations on the documents in the index. You can define as many indexes as you want in a single Elasticsearch cluster.**

## type

**A logical partition of an index. You can define one or more types in an index. Typically, a type is defined as a document that has a common set of fields.**

## mapping

**A process that defines document fields and their types as well as other index-wide settings. A mapping is similar to a schema definition in a relational database. Each index has a mapping. A mapping can either be defined in advance or automatically generated when you store a document for the first time.**

## document

**A JSON-formatted string which is stored in Elasticsearch, similar to a row in a relational database. Each document has a type and an ID. A document is a JSON object which contains zero or more fields, or key-value pairs.**

## field

**A simple value or a nested structure. Fields are similar to columns in relational database tables. Each field has a field type.**

## 23.3.7.2.2 How Elasticsearch on MaxCompute works

### *23.3.7.2.2.1 Overview*

Elasticsearch on MaxCompute is based on the open source Elasticsearch. It can run the Elasticsearch service on MaxCompute clusters.

On the MaxCompute client, you can start and manage your Elasticsearch service as needed and configure the number of nodes, disk space, memory size, and custom settings. The resources consumed by the Elasticsearch service are counted against your MaxCompute quota.

The following sections describe how Elasticsearch on MaxCompute functions work.

### *23.3.7.2.2.2 How distributed architecture works*

#### Basic principles

An Elasticsearch cluster consists of multiple nodes. MaxCompute ensures high availability by controlling the start and stop of Elasticsearch services and nodes, allocating computing resources, and implementing failover based on a centralized scheduling mechanism.

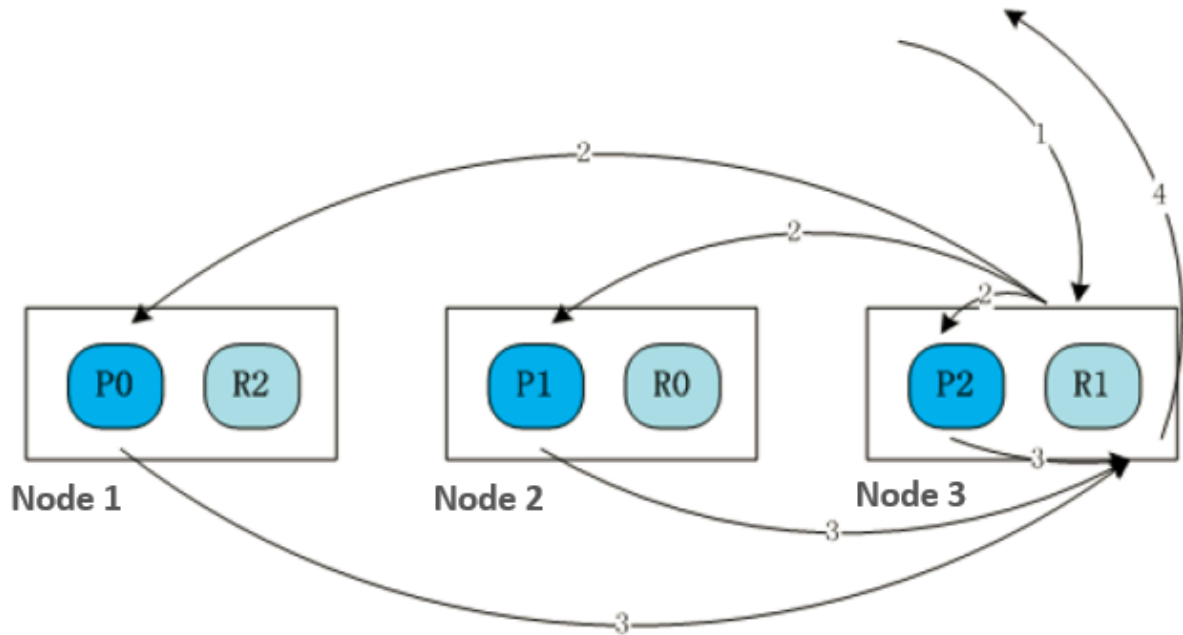
Data is replicated into multiple copies and stored in Apsara Distributed File System . This guarantees that no data is lost due to the failure of a few nodes.

An index is split into multiple shards, which are evenly distributed across multiple nodes in a cluster. The system simultaneously retrieves data shards in multiple nodes, improving retrieval performance.

## Implementation process

The following figure shows the distributed retrieval workflow.

Figure 23-9: Distributed retrieval workflow



As shown in the preceding figure, each cluster consists of three nodes. The index has three shards: P0, P1, and P2. These shards are distributed across the three nodes. Each shard is replicated in 1:1 mode, generating three replicas: R0, R1, and R2. The retrieval process is as follows:

1. A user sends a retrieval request to Node 3.
2. After receiving the request, Node 3 sends a retrieval request (2) to P0, P1, and P2 based on the recorded index shard information.
3. The nodes where P0, P1, and P2 are located search for the requested information in the specified shards. A retrieval result message (3) is sent to Node 3.
4. Node 3 collects the retrieval results from other nodes and returns the retrieval results to the user in an acknowledgment message (4).

When multiple nodes are performing data retrieval at the same time, the retrieval speed is improved. The performance of distributed retrieval increases with the number of nodes.

### 23.3.7.2.2.3 *How full-text retrieval works*

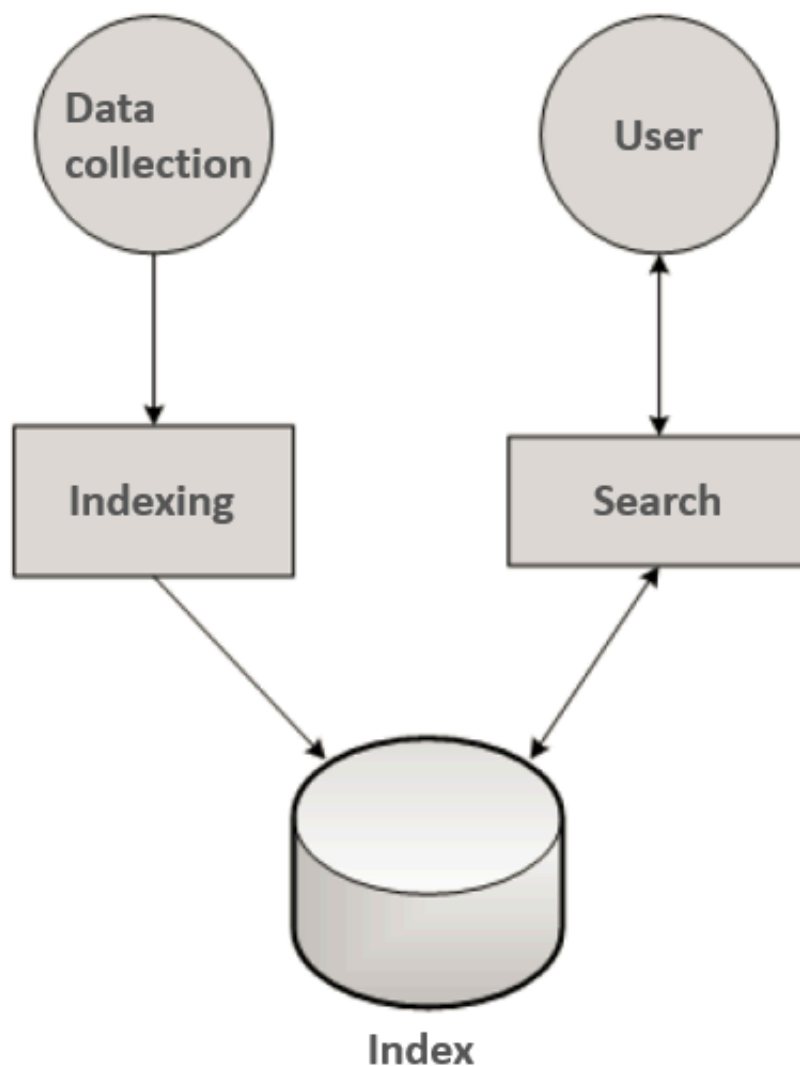
#### Basic principles

**Full-text retrieval refers to techniques used to search for data records containing specified contents from large volumes of texts. In the retrieval process, data in texts is segmented by words, and an inverted index is created based on mappings from words to documents to allow fast document retrieval.**

#### Implementation process

**The following figure shows the full-text retrieval process.**

Figure 23-10: Full-text retrieval process



**The retrieval process is as follows:**

1. The data collection module collects structured and unstructured data, converts the data into the field + value format, and submits the data to the indexing module.
2. The indexing module segments the data, creates inverted indexes based on a predefined indexing method, and saves the indexes. The field type, indexing method, and segmentation rules are configured on the retrieval management page.
3. The search module receives and processes user requests. Requests are parsed to obtain indexes, fields, and query statements, and then matched to records in the inverted indexes.
4. The indexing module returns data that meets user-defined requirements such as sorting rules and request quantity.

#### ***23.3.7.2.2.4 How authentication control works***

##### Basic principles

**Authentication control is implemented at the entrance used for external services to check whether users have been authorized to access the index libraries.**

##### Implementation process

**The authentication control process is as follows:**

1. Elasticsearch on MaxCompute provides retrieval management and O&M platforms that are only accessible after logon. User account information is verified and authenticated by a centralized authentication module before logon. Any user who fails the authentication is denied access to the platforms.
2. The administrator can use the MaxCompute client to add Elasticsearch users and configure permissions for the users.
3. The system authenticates all users who attempt to access index libraries. After passing authentication, you will be able to retrieve or perform operations on data in the libraries.

## 24 DataWorks

### 24.1 What is DataWorks?

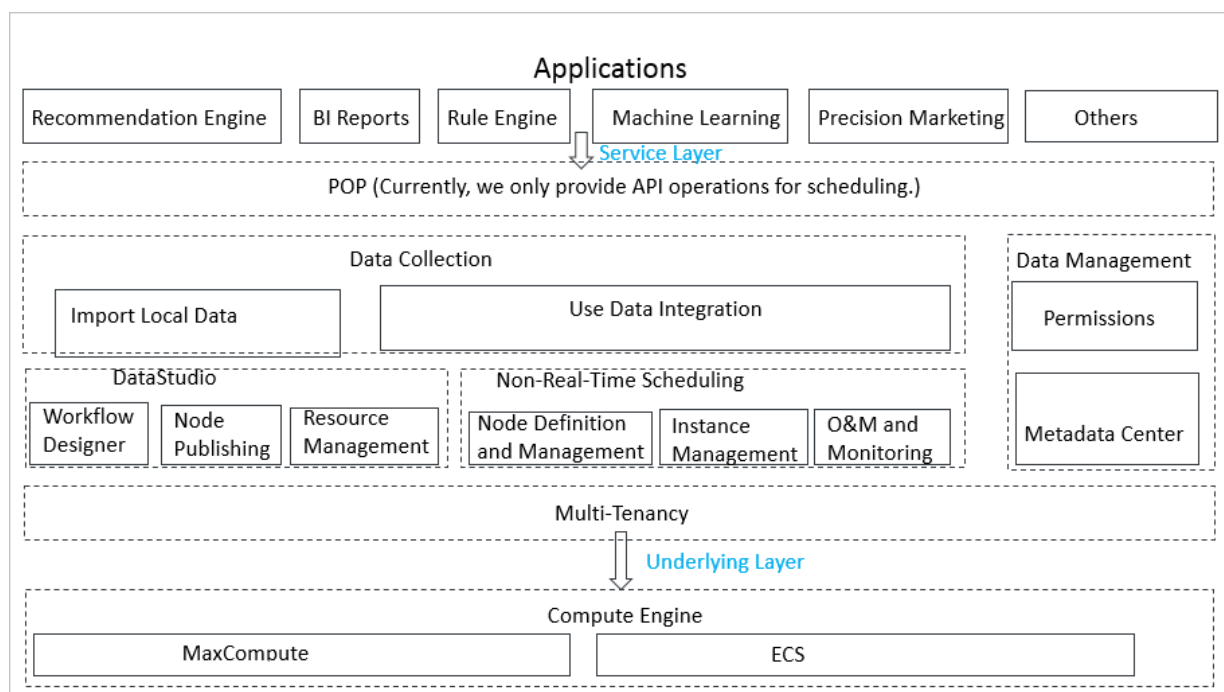
#### 24.1.1 Product overview

**DataWorks is a big data platform provided by Alibaba Cloud. It provides end-to-end solutions for enterprises and individual users, and is integrated with the development platform, the management system, and the offline scheduling system.**

**DataWorks is aimed at mining the full value of the data.**

- **It enables large enterprises to build petabyte-level and even exabyte-level data warehouses. The enterprises can improve their business operations through data integration, data asset management, and data analysis.**
- **Small and medium-sized enterprises and individual users can quickly build data-based applications, which drive data based innovations.**

Figure 24-1: Product components



**DataWorks consists of an integrated development environment (IDE), a scheduling system, a data integration tool, and a data management system.**



- **Integrated development environment (IDE):** A development tool that can be used to write SQL, MapReduce (MR), or shell code. The IDE supports collaborative development and version control. By using the visual process design tool, you can quickly define the dependencies among different tasks.
- **Scheduling system:** A system that can schedule millions of tasks in a day. You can manage your tasks online, and view the logs, scheduling status, and monitoring alerts.
- **Data integration tool:** An integration tool that can be used to configure synchronization tasks between heterogeneous data sources. More than 80% of databases and storage systems provided by Alibaba Cloud, include relational databases, FTP, HDFS, which can be configured as a source or a destination in the synchronization task. You can also create a task that runs periodically to synchronize data on a periodic basis.
- **Data management system:** A system that can be used to manage data in MaxCompute. You can manage permissions, view the data lineage, and view the metadata.

## 24.1.2 Scenarios

### Large data warehouses

Enterprises can use DataWorks in Apsara Stack to build large data warehouses.

DataWorks provides superior data processing capabilities:

- **Massive storage:** Supports PB- and EB-level data warehouses and linear expansion of storage size.
- **Data integration:** Supports data synchronization and integration across heterogeneous data sources to eliminate data islands.
- **Data analytics:** Supports MaxCompute-based big data processing capabilities, 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 periodic task execution and processing for millions of tasks in a day, real-time task monitoring, and timely error alerting.

## Data-driven management

- **Innovative businesses:** Data mining, data modeling, and real-time decision making can be implemented based on big data analytics results provided by DataWorks.
- **Small and medium-sized enterprises:** With DataWorks, data can be quickly analyzed and put to commercial use, which help enterprises generate marketing strategies.

## 24.2 Benefits

### Capability of processing big data

DataWorks uses MaxCompute as its compute engine, which allows for a maximum of 5,000 servers in a single cluster. DataWorks can access data from different clusters, which allows you to easily process your big data. The scheduling system can run millions of nodes each day, and you can configure rules and alerts to monitor the running of nodes.

### Key features

- DataWorks supports join operations for trillions of data records, millions of concurrent jobs, and petabytes (PB) of I/O throughput each day.
- You can share data across clusters, and scale out clusters (a maximum of tens of thousands).
- DataWorks provides efficient and easy-to-use SQL and MR engines that support the majority of standard SQL syntax.
- MaxCompute (formerly known as ODPS) stores data in triplicate. It also adopts multiple access control mechanisms such as read/write request authentication, application sandboxing, and system sandboxing. All these mechanisms secure user data, and protect data against loss, leak, and compromise.

### Integrated data processing environment

DataWorks integrates development, scheduling, monitoring, and alerting for nodes, and management of data.

### Key features

- DataWorks provides you with all the required features for data processing.
- You can design workflows in a visual designer that is similar to Kettle.

- **DataWorks provides an online collaborative development environment. You can create and assign roles for varying tasks, such as development, online scheduling, maintenance, and data permission management, without locally processing data and tasks.**

#### Integration from heterogeneous data stores

**DataWorks supports read/write functions for various types of data stores. You can configure dirty data filtering and bandwidth throttling.**

#### Key features

- **DataWorks can read data from data stores of the following types: MySQL, Oracle, PostgreSQL, RDS, MaxCompute, FTP, OSS, HDFS, Dameng, and Sybase.**
- **DataWorks can write data to data stores of the following types: MySQL, Oracle, PostgreSQL, RDS, MaxCompute, Memcache, OSS, HDFS, Dameng, and Sybase.**
- **You can configure dirty data filtering and bandwidth throttling.**
- **DataWorks supports recurring nodes, including recurring data synchronization nodes.**

#### Web-based software

**You can use DataWorks whenever an internal or public network is available.**

#### Multi-tenancy

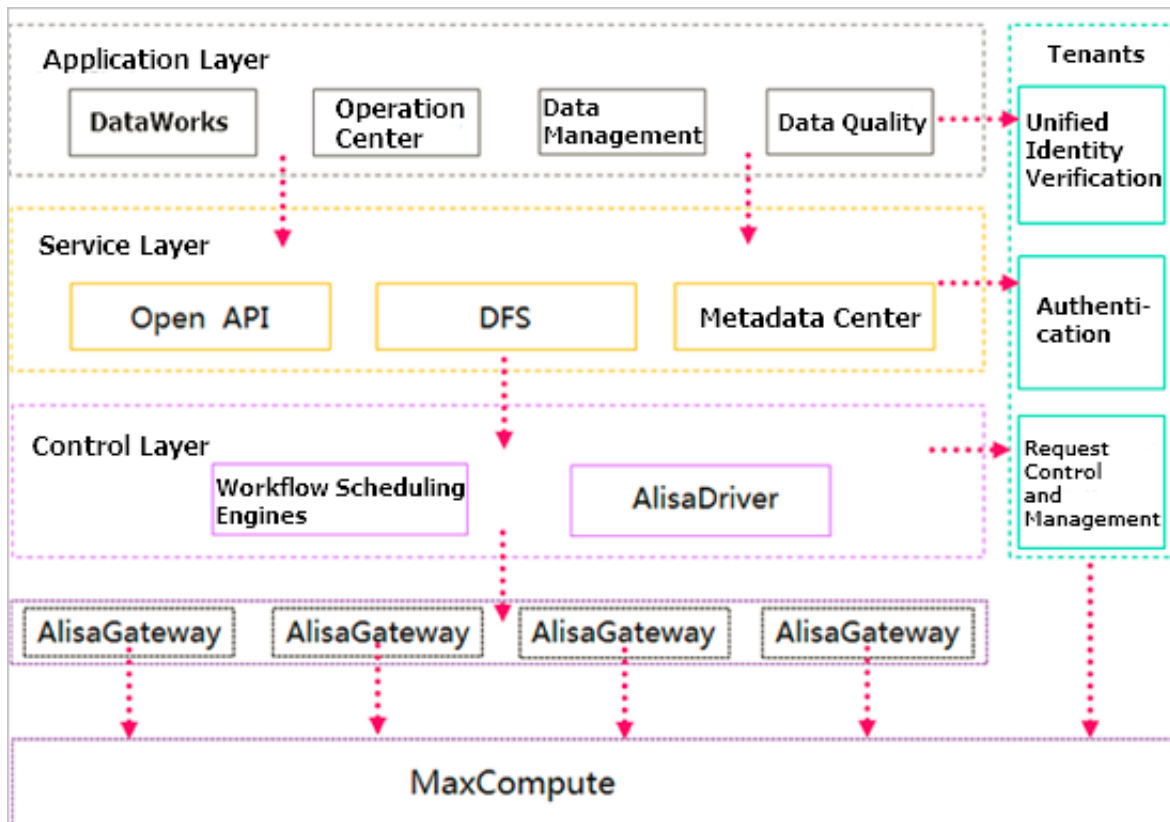
**DataWorks adopts multi-tenancy to isolate data among tenants. Each tenant separately manages their own permissions, data, resources, and members.**

#### Open platform

**DataWorks provides all modules as components and services. You can use APIs to develop extra functions of DataWorks.**

## 24.3 Architecture

### System architecture



DataWorks adopts the design of components and services, and consists of the following three layers.

- **Control layer:** the core of DataWorks batch data processing. The workflow scheduling engine generates and runs node instances. AlisaDriver coordinates and controls the running of all nodes.
- **Service layer:** provides services for the application layer and other external applications.
- **Application layer:** runs on top of the service layer, and provides the graphical interface for user interactions.

### Security architecture

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.**

#### Multi-tenancy

**DataWorks adopts multi-tenancy.**

- **Storage and compute resources are scalable. You can manage your own resources, and request resource quotas as needed.**
- **Tenants are isolated. Each tenant separately manages its own data, permissions, accounts, and roles.**

## 24.4 Services

### 24.4.1 DataStudio

**DataStudio is an integrated development environment (IDE) in DataWorks, which supports database warehousing, data query, ETL, and algorithm development for big data. It also supports online collaborative development and version control.**

#### Features

- **DataStudio provides a visualized workflow designer, which is similar to the Kettle tool. It allows you to design workflows, and manage nodes of each workflow.**
- **DataStudio supports the upload of local files.**
- **DataStudio supports data integration from heterogeneous data stores.**



**Note:**

**Data Integration supports the following data store types:**

- **Types of source data stores: MySQL, Oracle, PostgreSQL, RDS, MaxCompute, FTP, OSS, HDFS, Dameng, and Sybase.**
- **Types of target data stores: MySQL, Oracle, PostgreSQL, RDS, MaxCompute, Memcache, OSS, HDFS, Dameng, and Sybase.**

- DataStudio provides a web-based programming and debugging environment that allows you to create SQL, MR, shell (limited support), and data synchronization nodes.
- DataStudio supports node deployment across MaxCompute projects. You can deploy nodes and code to the scheduling system across different workspaces.
- DataStudio adopts version control, node locking, and conflict detection mechanisms to facilitate collaborative development.
- DataStudio enables you to search and use MaxCompute tables, resources, and user-defined functions.

## 24.4.2 Data Management

Data Management enables you to perform queries on the tables, view details of tables, and manage permissions on tables. You can also add tables to your favorites. For more information, see the Data Management topic in *DataWorks User Guide*.

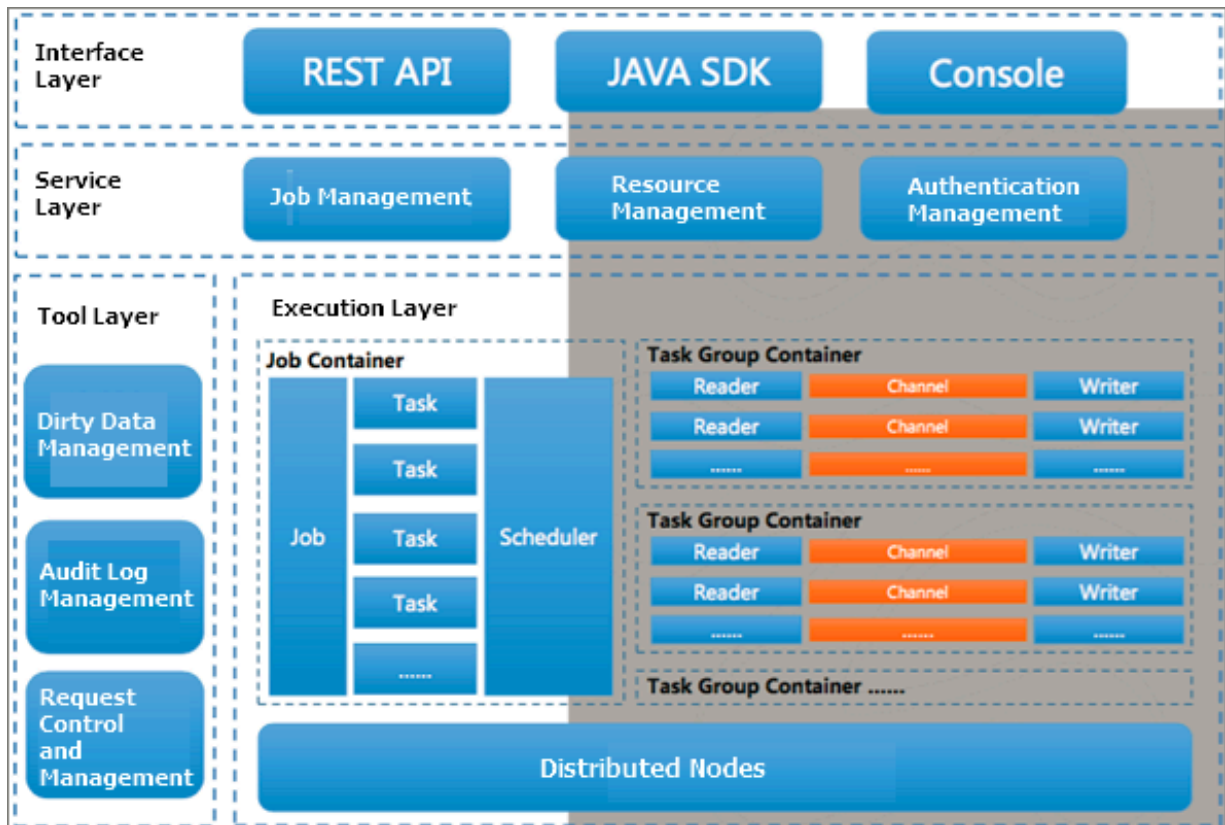
## 24.4.3 Data Integration

Data Integration is a data synchronization platform that provides stable, efficient, and scalable services. Data Integration provides transmission channels for batch data stored in MaxCompute, and Realtime Compute. Data Integration implements fast integration on data from heterogeneous data stores.

Data Integration provides connectors and a framework. The connectors are used for reading and writing data, and the framework is used for common operations in data synchronization and transmission. Data Integration provides two types of connectors:

- Reader: reads data from the data store
- Writer: writes data to the data store

You can develop readers and writers for Data Integration to support more data store types.



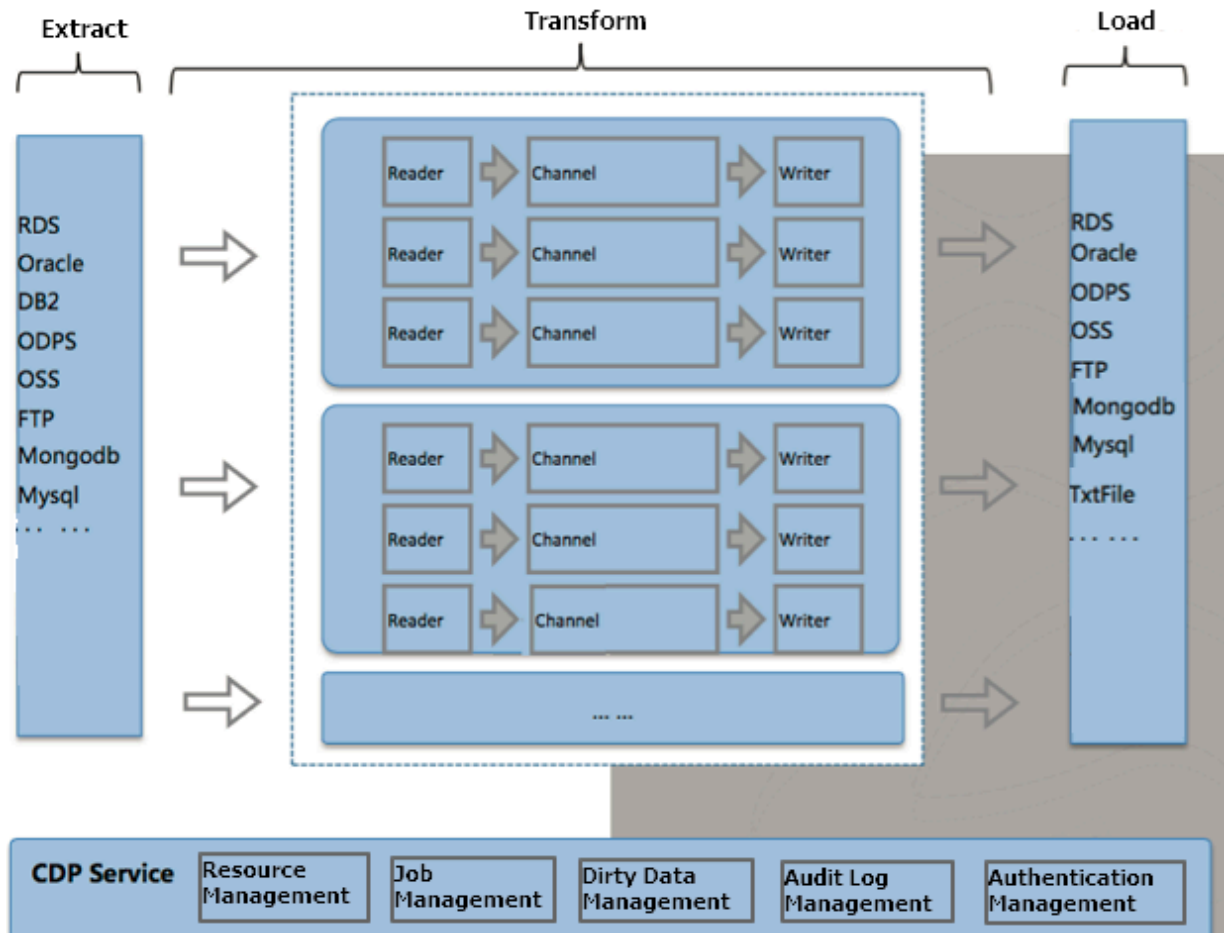
The interface layer provides three methods of using the Data Integration service: RESTful API, Java SDK, and console. RESTful API supports multiple programming languages. We recommend that you use Java SDK to avoid manual operations such as signature, authentication, and HTTP request making. The console is developed based on the command line tool, which allows you to use the majority of Data Integration functionalities. Data Integration also provides developers with a web UI based on RESTful API.

The service layer includes resource management, job management, and authentication management. For more information, see the product overview.

The tool and execution layers form the core of Data Integration. The two layers run extract, transform, load (ETL) jobs. All synchronization jobs that are submitted to

Data Integration run on the execution layer. The execution layer uses DataX as the synchronization engine.

Figure 24-2: Process of an ETL job





## Features

- **Data Integration supports the following types of data stores:**
  - **Relational databases:** MySQL, PostgreSQL, Oracle, Db2, and general relational databases
  - **NoSQL databases:** Table Store and Memcache
  - **Big data platforms:** MaxCompute
  - **Unstructured data stores:** OSS, HDFS, and FTP

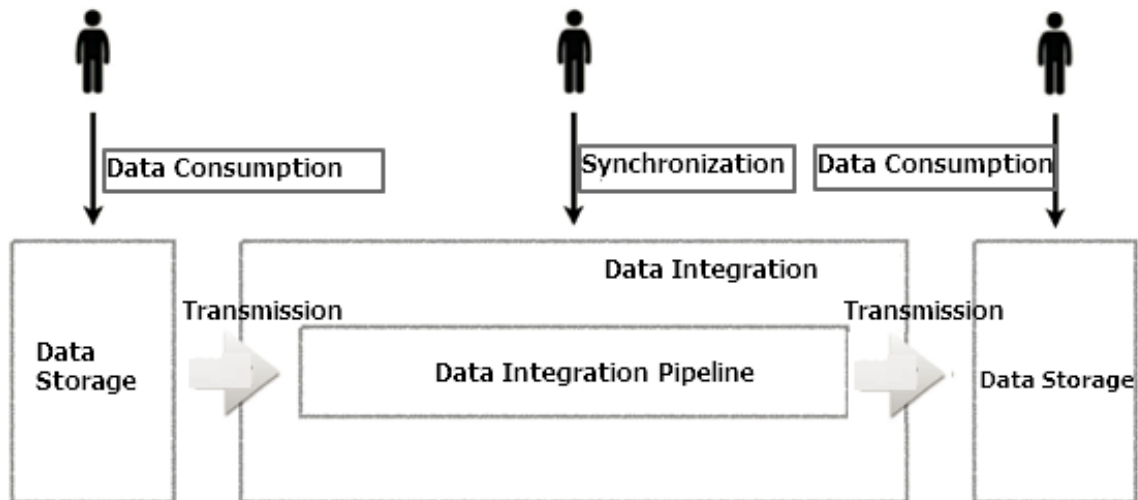
**You can use the following JDBC URLs when you configure connections to general relational databases such as Dameng, Db2, and PPAS:**

- **Dameng:** jdbc:dm://ip:port/database
- **Db2:** jdbc:db2://ip:port/database
- **PPAS:** jdbc:edb://ip:port/database

**Data Integration supports recurring batch synchronization from the source to the target. For example, you can configure a data synchronization node that runs on a daily, weekly, or monthly basis. When the batch data synchronization node starts, a snapshot of source data is taken. The system then reads data from the snapshot and writes the data to the target. Each batch data synchronization node has a life cycle.**

**Data Integration only defines the synchronization process. Data transmission during the synchronization process is under the control of the Data Integration cluster. Data channels and streams are invisible to users. Data Integration does**

not provide any API for data consumption. You need to consume data on both the source and target data stores.



- **Consistent data quality**
  - Data Integration supports conversions between different data types.
  - It accurately identifies, filters, collects, and displays dirty data to ensure the quality of data.
  - Data Integration supports job performance reporting, which helps you track node status, such as data volume and dirty data.
- **Efficient data transmission**
  - Data Integration supports one-way data channels, and allows a single process to reach the maximum data transfer rate (up to 1,600 Mbit/s) on each server.
  - It adopts a distributed architecture, and supports transmission for gigabytes (GB) to terabytes (TB) of data.
- **User-friendly control experience**
  - Data Integration implements accurate control of channels, record streams, and byte streams.
  - You can rerun any threads, processes, and jobs that fail.

- **Clear core design**
  - **Data Integration provides a professional framework and an efficient execution engine. The engine supports common connectors, standardizes the process of developing connectors, and automatically detects new connectors.**
  - **Data Integration provides clearly defined and easy-to-use connector APIs that allow developers to focus on the implementation of the connector instead of the framework.**

## 24.4.4 Tenant management

- **Workspace management**

**The Workspace Management page includes basic workspace settings.**

- **Sandbox whitelist: IP addresses and domains that can access the workspace**
- **Compute engine: information of the MaxCompute engine**

- **User management**

**On the Members page, you can assign and unassign a role from specified members.**

- **Permission list**

**On the Permissions page, you can view the permissions of a role for tables and workspaces.**

## 24.4.5 Data Quality

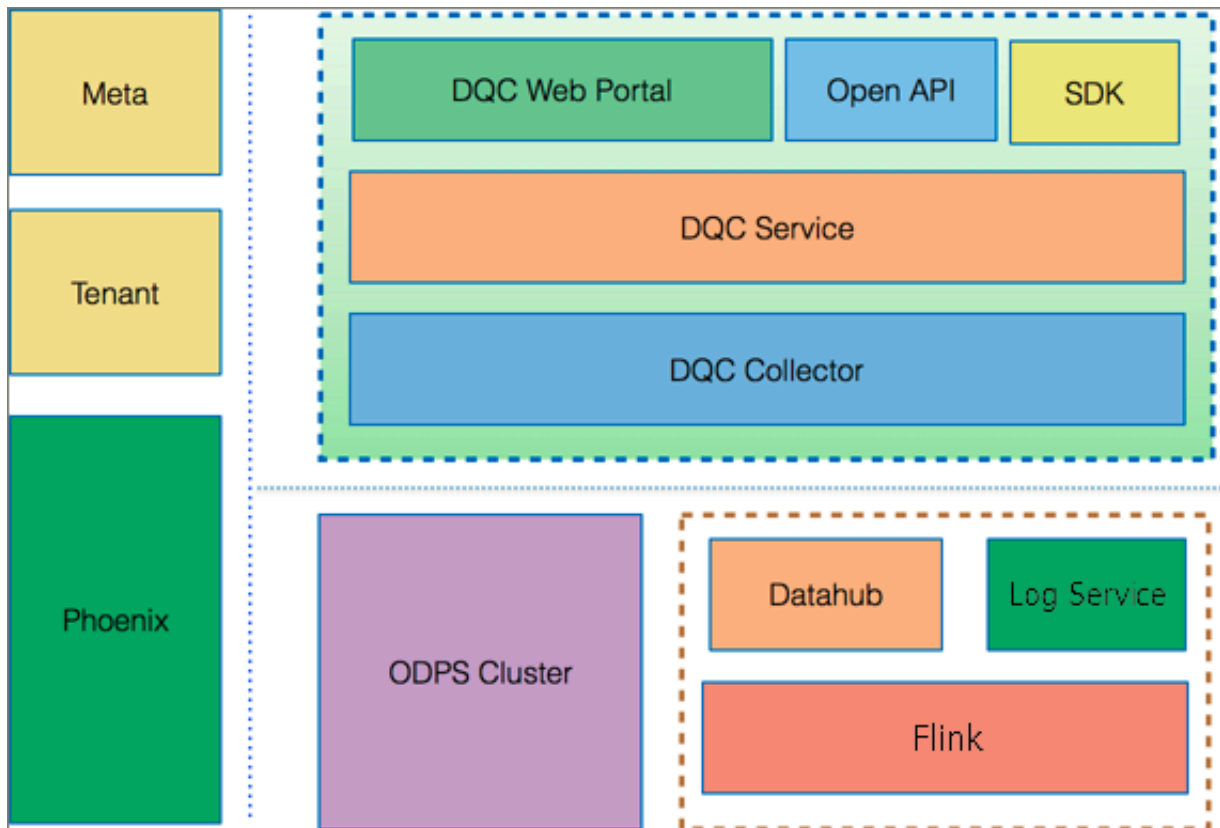
### 24.4.5.1 Overview of Data Quality

**Data Quality is a platform that provides data quality check and management services. You can use it to monitor both real-time and batch data during the entire data processing cycle. When you use Data Quality to monitor real-time data, it can detect discontinuity, delay, and other user-defined data issues in DataHub data streams. When you use Data Quality to monitor batch data, it can detect abnormal data in the production process, protect downstream data from being affected by abnormal data, and promptly notify you about the abnormal data. This helps to ensure the correctness of your data.**

**Data Quality requires the access to the metadata, fields, and tables, and requires user and tenant management. In the scenario of monitoring batch data, Data Quality uses MaxCompute as the compute engine. In the scenario of monitoring**

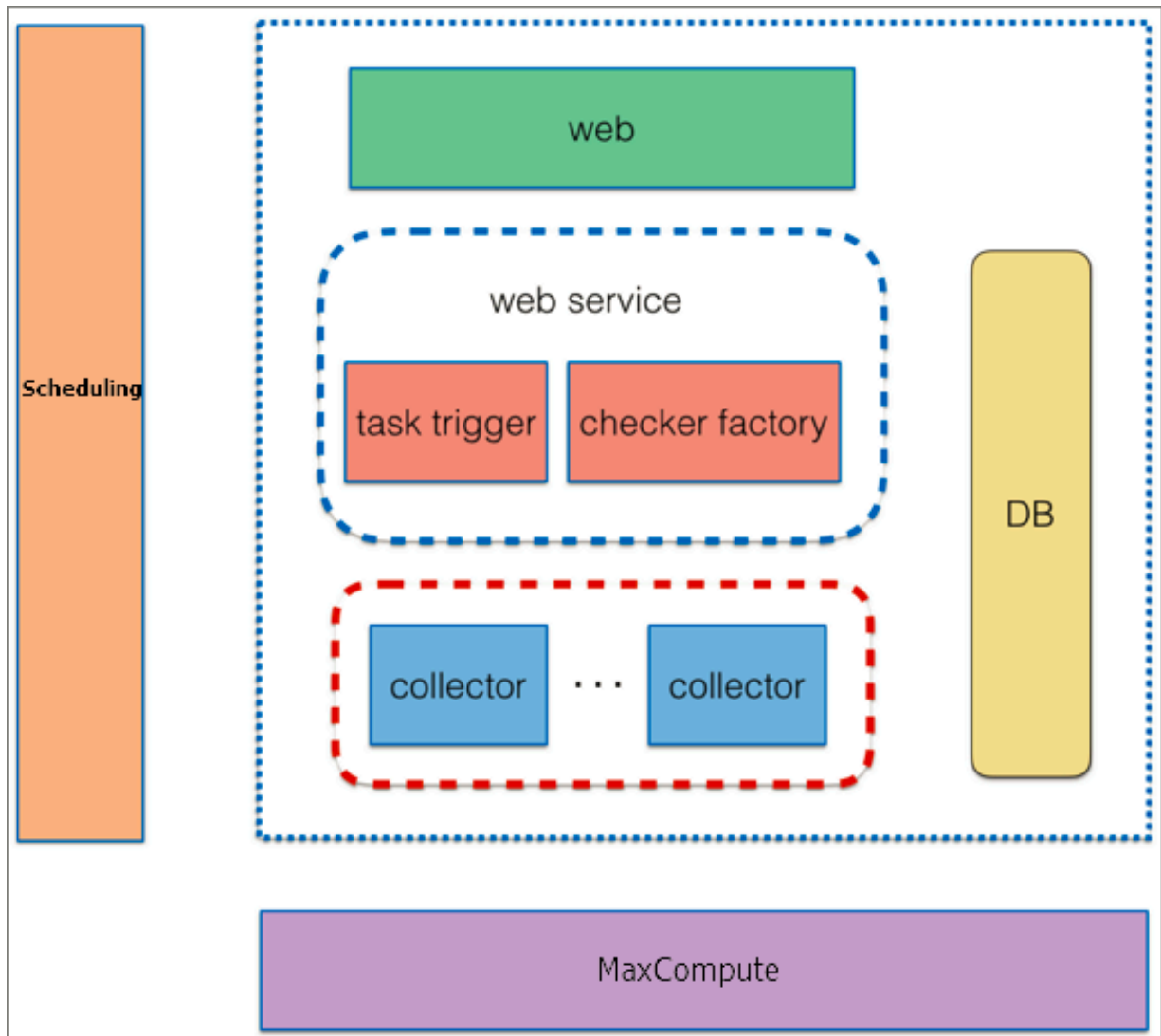
real-time data, Data Quality uses the Flink framework as the streaming data processing tool. Data Quality consists of three components: the web portal, the check service, and the data collection service.

Figure 24-3: Data Quality architecture



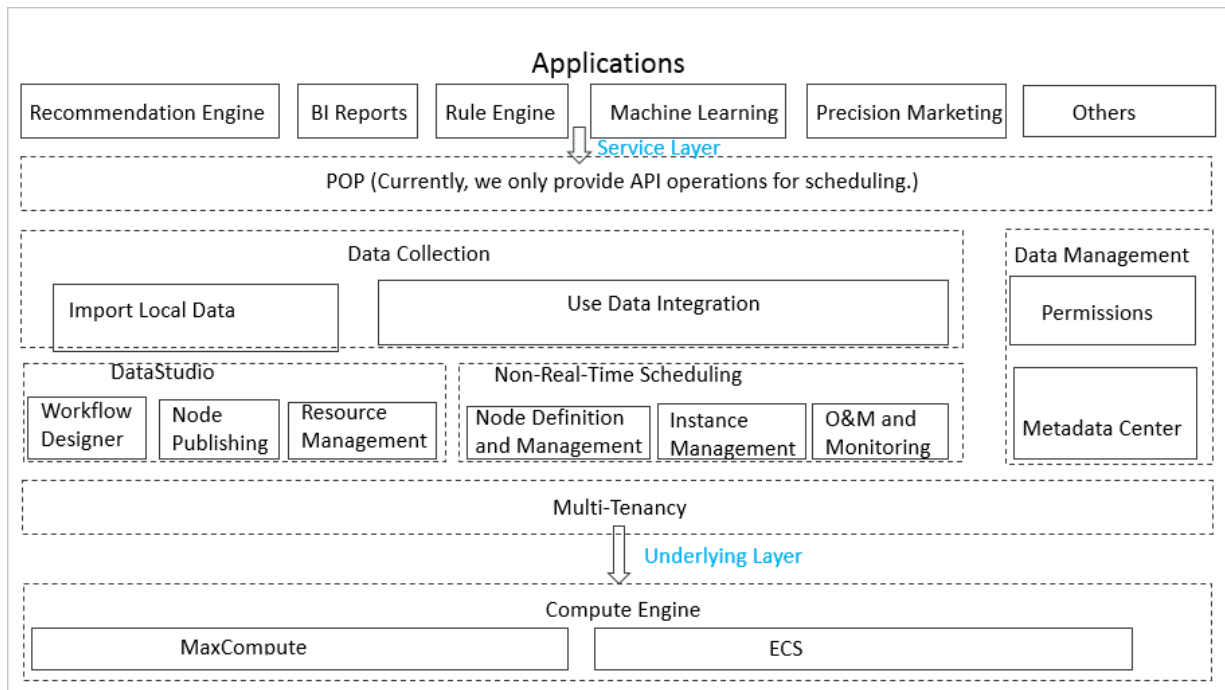
## 24.4.5.2 Use Data Quality to monitor batch data

Architecture



- **Web:** Web UI provides a graphical interface for users. It consists of rule management, search by node, subscription management, dashboard, permission control, and cache management.
- **Web service:** The web service layer provides access to databases, checks data quality, parses jobs, and triggers jobs. The checker factory module checks samples by using quality check logics such as comparison of fixed value, fluctuation, and variance detection.
- **Collector:** The collector module consists of multiple data collection engines that obtain data samples based on user specified rules. Data collection engines classify the rules based on potency, rule types, and sampling methods. Before sending the rules to MaxCompute to obtain data samples, data collection engines apply logical splitting and combination to the rules.

## Principle



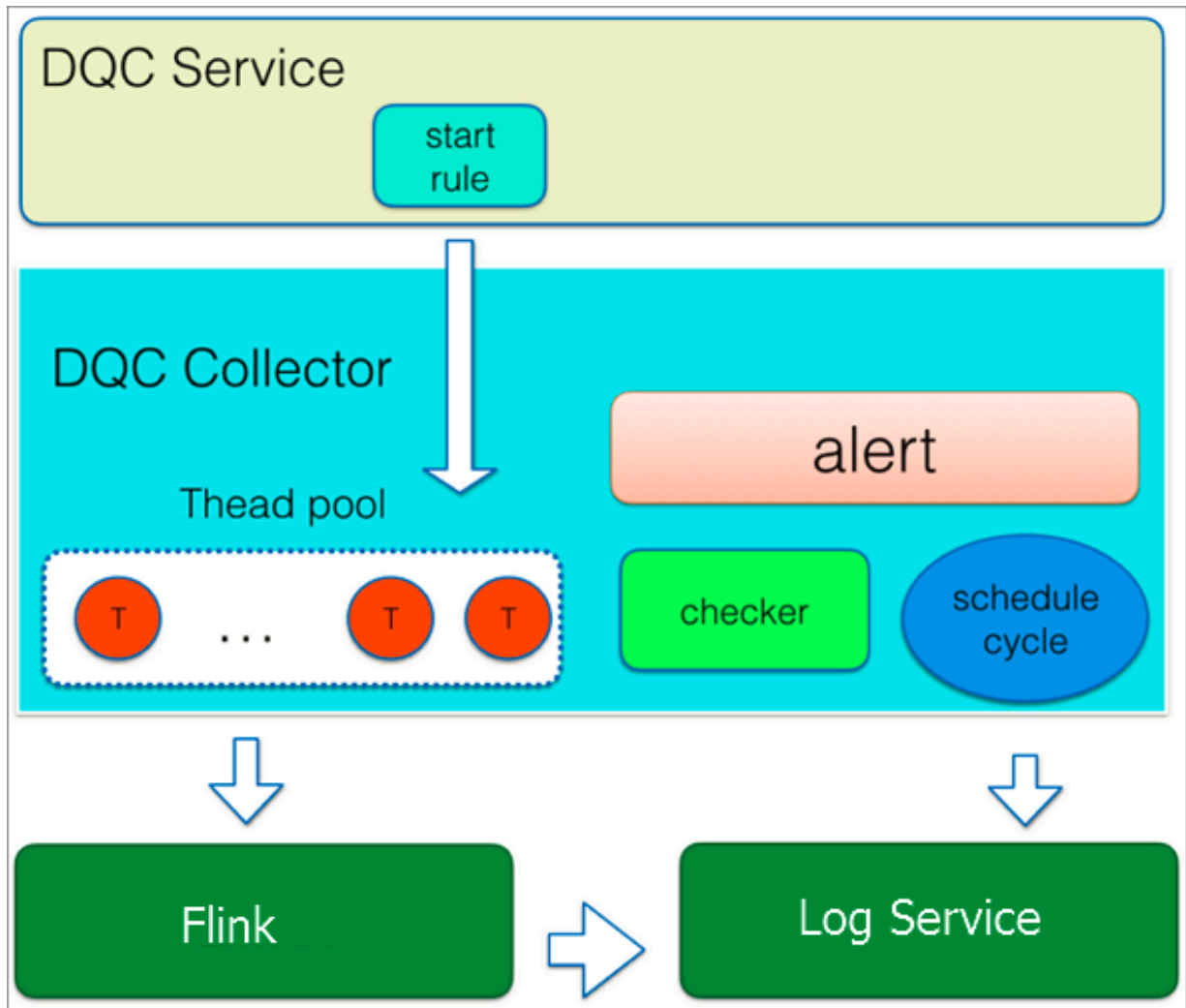
1. The scheduling system sends a request that triggers the service module to check the quality of data in the specified partitions of a table. The request contains the partition expression, table information, and node schedule.
2. Based on the partition expression, a server in the service module obtains the set of rules that are applied to the current node. The server submits a request for obtaining data samples to data collection engines and returns the request result to the scheduling system. The scheduling system first allocates resources to run nodes that are associated with strong rules.
3. The data collection engines further classify the set of rules based on potency, rule types, and sampling methods. The MaxCompute cluster collects data samples based on the sampling methods.
4. After finishing data sampling based on strong rules, the data collection engines instruct the service module to check data quality. After the quality check, the service module sends the check results to the scheduling system, and the scheduling system determines whether to block the node.
5. After the quality check by using strong rules, the service module returns the results to the data collection engines. The data collection engines continue the sampling process, and send the processed data for check based on weak rules. After the weak rule check is complete, the quality check ends.

## Benefits

- **Data Quality provides built-in rule templates and comprehensive data quality metrics. The templates support field and table level rules with a fluctuation threshold or fixed value comparison. You can create rules from the templates to check whether data entries are null or unique or use discrete values, the maximum, minimum, average, or sum to evaluate the data quality. You can also create custom rules for special requirements.**
- **Data Quality clusters are horizontally scalable. You can add servers if Data Quality reaches the maximum concurrency. Data Quality also includes a reliable fault-tolerance system that ensures that data collection jobs are accurate and consistent.**
- **Data Quality supports rule classification based on potency and severity levels . When you use Data Quality to monitor batch data, you can classify rules into weak and strong rules based on potency. You can also set thresholds to reflect the warning and error severity levels of check results based on the deviation from the expected value. When strong rule check results show a significant deviation from expected values, the node is blocked to protect downstream data against dirty data. This ensures the correctness of data during the data processing cycle.**
- **Data Quality provides a potency based execution mechanism that first runs the nodes that are associated with strong rules. The collector module supports running nodes based on the potency.**
  - **If available resources are limited, this mechanism ensures that you first run nodes that are associated with strong rules.**
  - **If available resources are sufficient, this mechanism allows nodes that are associated with weak rules to run.**

### 24.4.5.3 Use Data Quality to monitor real-time data

#### Architecture



Rules for monitoring real-time data are converted into Flink SQL statements. Data Quality uses Flink to read data from DataHub and write check results to Log Service. The collector module of Data Quality regularly obtains abnormal data from Log Service, writes the data to Redis, and then triggers alerts. The service module of Data Quality synchronizes the alerts from Redis to other databases for users to query.

#### Principle

1. After you enable a rule, the service module creates a Logstore. The service module uses an SQL parser to declare a dimension table used for referencing a DataHub topic. The service module uses a rule converter to generate a CREATE TABLE statement and combine table operations. Then, the service module submits a Flink job and updates the next quality check time.



2. One of the servers in the service module first establishes a lock to serve as the master. The master collects data from DataHub topics on a regular basis and sends the data to the collector module for quality check.
3. The collector module uses a LogHub consumer to subscribe to the Logstore . Then, the collector module writes abnormal data to Redis, and determines whether to send alerts.
4. The service module starts the Quartz scheduler worker service, and writes the data from Redis to another database for users to query.

#### Benefits

- You can use Data Quality to monitor real-time data in various scenarios. Data Quality detects discontinuity and delay of real-time data streams, and allows you to create Flink SQL queries as custom rules. Data Quality also supports join operations for multiple streams and dimension tables.
- Data Quality supports monitoring on data delay at the level of seconds.
- Data Quality allows you to specify an alert interval and the number of alerts for each rule to reduce redundant alerts.
- Data Quality allows you to set thresholds at the warning and error severity levels . This helps you identify the deviation of check results from expected values.
- Data Quality uses hashing algorithms to remove duplicate alerts. This ensures data idempotence in the real-time computing process.

### 24.4.6 Data Asset Management

Data Asset Management provides portal management, data asset category management, data source management, and business unit management. Using the Data Asset Management service helps you understand your core data assets and standardize your management process.

### 24.4.7 Real-Time Analysis

The Real-Time Analysis service, which is based on MaxCompute, provides you with quick data query and data preview. The service is suitable for data analysis and data exploration.

- Supports creating, renaming, and deleting folders and files.
  1. Click Run to run the SQL statements.
  2. View the results.

## 24.4.8 Data Service

Data Service provides features such as API hosting, authentication, authorization, and management. You can create APIs for tables, and publish the APIs by using the API Gateway service.

**Features:**

- Supports various data sources, including relational databases, and NoSQL databases.

Supported data sources: MySQL, Oracle, PostgreSQL, ApsaraDB for RDS, Table Store, MongoDB, and Lightning.

- Provides the wizard mode, which can be used to create APIs without writing code.
- Provides the script mode. You can create APIs by writing SQL statements.
- Provides accurate access control. You can customize permissions on APIs, table rows, and table columns.
- Provides API Gateway and HTTP request methods.
- Supports a variety of network environments, including local private networks, VPCs, and Alibaba Classic networks.
- Provides API creation, grouping, and publishing.
- Provides organization-based API isolation.
- Provides Open API, which supports registering, managing, and testing APIs.
- Supports a variety of API execution environments, including stand-alone environments and the EAS container service.
- Supports debugging APIs online. You can view the API call information and the performance in real time.

## 24.4.9 Intelligent Monitor

Intelligent Monitor is a system that monitors and analyzes tasks in DataWorks. Intelligent Monitor sends alerts based on specified rules, times, methods, and recipients. It automatically uses the most appropriate alert time, method, and recipients. Intelligent Monitor has the following benefits:

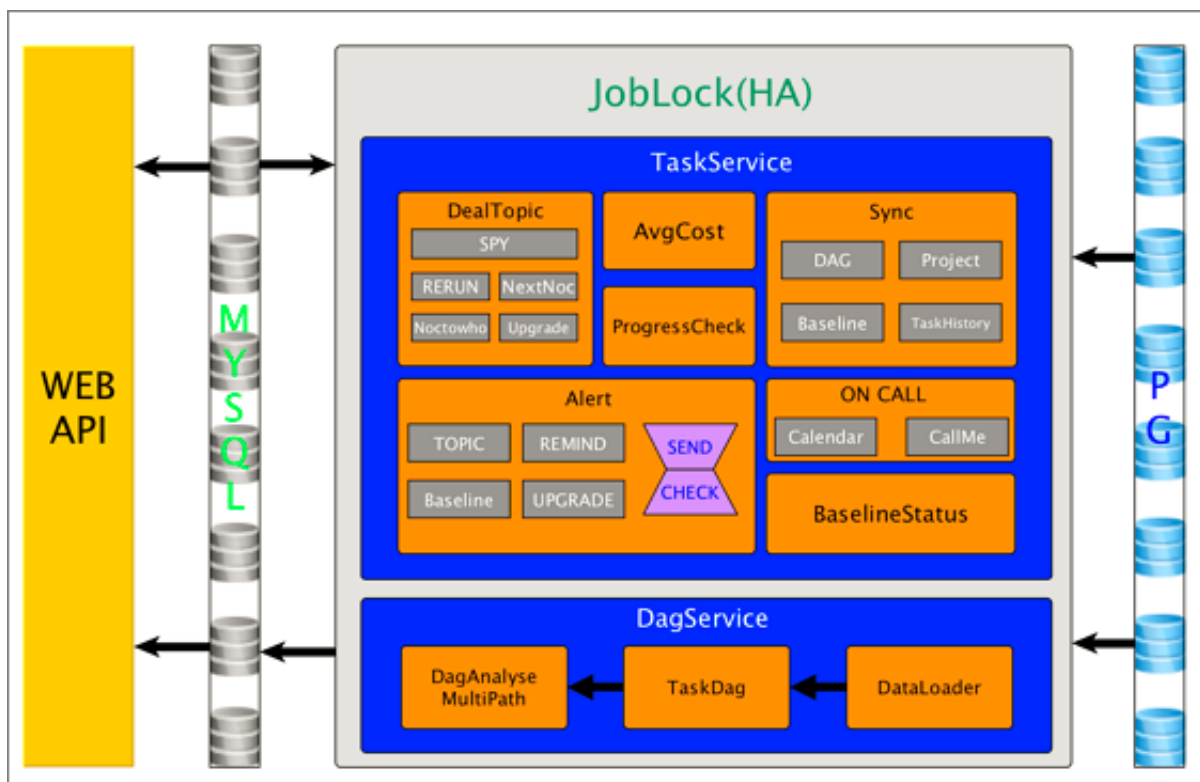
- Alerting rules are easy to configure.
- You are free of invalid alerts.
- All important tasks are monitored.

General monitoring systems cannot meet the requirement of DataWorks. The reasons are:

- DataWorks has considerable tasks, and you cannot accurately locate all tasks that need to be monitored. Dependencies between tasks are complex. Even if you know what are the most important tasks, you have difficulties in figuring and monitoring all related tasks. If you monitor all tasks, invalid alerts may be generated. You have to determine which alerts are useful.
- Different tasks require different alert configurations. Some alerts are sent when the task runs for more than one hour, while others are sent when the task runs for more than two hours. It is difficult to specify different configuration for each single task.
- Different types of alerts are sent at different time. For example, an alert for an unimportant task can be send after you begin to work, and an alert for an important task should be sent immediately when the error occurs. Moreover, the importance of each task is hard to determine.
- Different alerts require different operations to turn off.

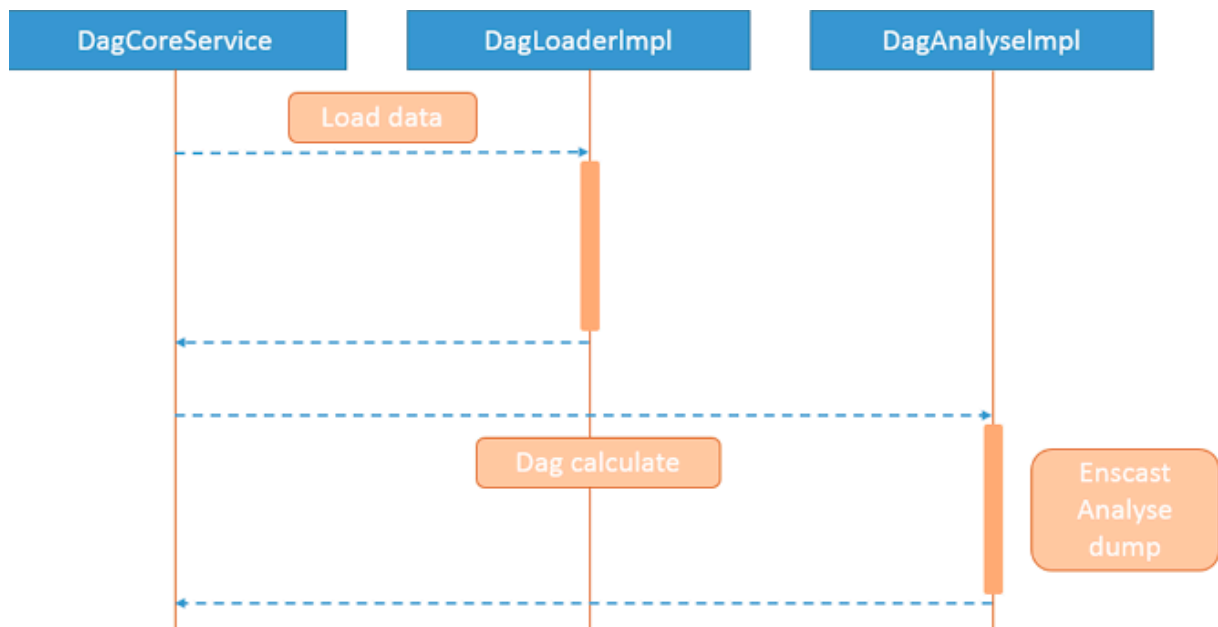
The Monitor feature automatically creates alert rules for tasks that are configured with baselines. You can also customize alert rules by completing basic settings.

Technical architecture



- **Dagservice:** Analyzes all tasks on each DAG based on the baseline settings. Dagservice determines the estimated completion time, the key path, the required completion time, and whether the task is suspended. The information collected by Dagservice provides the basis for TaskService.
- **TaskService:** Performs different tasks based on the information provided by DagService, including estimating the completion time, acquiring and fixing events, and customizing baseline alerts.
- **WebService:** Provides HTTP APIs that can be called to send requests. You can call APIs to view the Intelligent Monitor information, such as baseline instances, alert information, events, and gantt charts.

How it works



DagService collects the information of all nodes on each DAG based on the baselines and the average running time of each task. The information contains the estimated completion time, the required completion time, the key path, whether the node is blocked, and whether the node is a child of a suspended node.

TaskService runs tasks based on the task configuration and the information provided by DagService. The database lock ensures that one task is executed by only one server. When a server is down, another server takes over the task, which ensures the high availability of the monitoring service.

## 24.4.10 Scheduling system

### 24.4.10.1 Overview

The scheduling system is one of the core systems in DataWorks, which is responsible for scheduling all tasks. The scheduling system runs tasks based on the specified time and the dependencies.

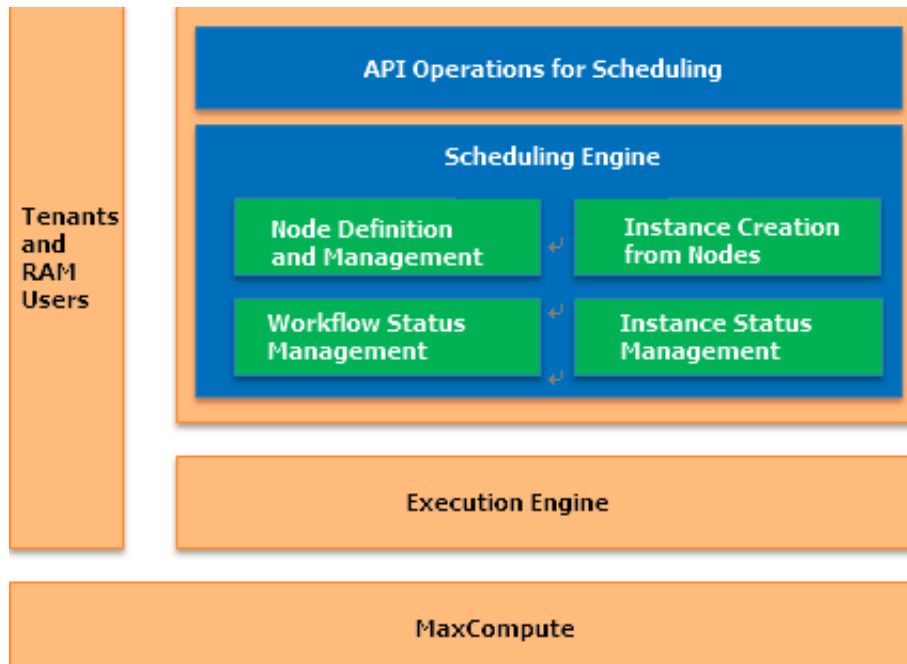
- The scheduling system can schedule millions of jobs.
- The distributed execution architecture enables linear scalability of concurrent jobs.
- You can configure scheduling tasks at custom intervals in minutes, days, hours, weeks, or months.
- You can configure the dependencies of different tasks. You can choose from general dependency, cross-period dependency, or self-dependency.
- You can dry run a task or suspend a task.
- You can create and run an ad-hoc workflow.
- You can view the workflow in a directed acyclic graph (DAG), which provides you with a clear view while troubleshooting.
- You can monitor tasks in real time, and send alerts by sending SMS messages or emails.
- You can rerun tasks, terminate processes, set the status of tasks to Successful, or suspend tasks.
- You can create retroactive tasks. Instances in different periods run serially.
- You can view the total number of tasks, the number of task errors, the number of scheduling tasks, the top 10 scheduling tasks that consume the most resources, the top 10 scheduling tasks that take the longest to run, and the distribution of task types.

### 24.4.10.2 Concepts

- **Node:** A node represents a task in the scheduling system. Node properties include the node type, the code version, the specified task start time, and the dependencies between tasks.
- **Instance:** An instance is created whenever a task (node) runs. The instance has all properties that the task (node) has. In addition, the instance contains the runtime information such as the instance status and the time when the status changes.

- **Workflow:** A workflow is composed of several interdependent instances. The scheduling system consolidates all instances in a day into a workflow for unified management. A workflow has its own status, which is determined by the status of each instance in the workflow.

### 24.4.10.3 Architecture



The preceding figure shows the architecture of the scheduling system and its relationship with other systems.

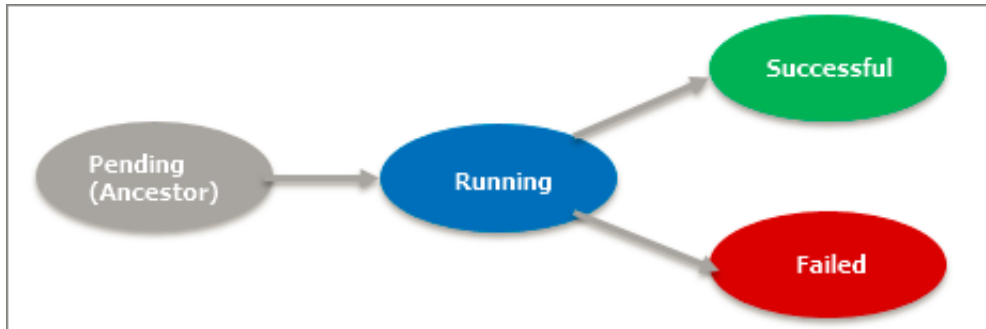
The scheduling engine is the core of the scheduling system. It contains four main modules.

- The node definition and management module maintains node definitions submitted by users, including the code, the specified task start time, and the dependencies. An instance is generated from the node configurations at a fixed time every day.
- The instance status management module manages the status changes after an instance runs.
- The workflow status management module maintains the status changes after a workflow runs. (A workflow is a set of instances with dependencies.)
- The scheduling system provides APIs for other systems to perform INSERT, DELETE, UPDATE, and SELECT operations.

The resources that are used by the scheduling system are isolated among tenants . Before a task instance runs, the scheduling system schedules the instance to the execution engine.

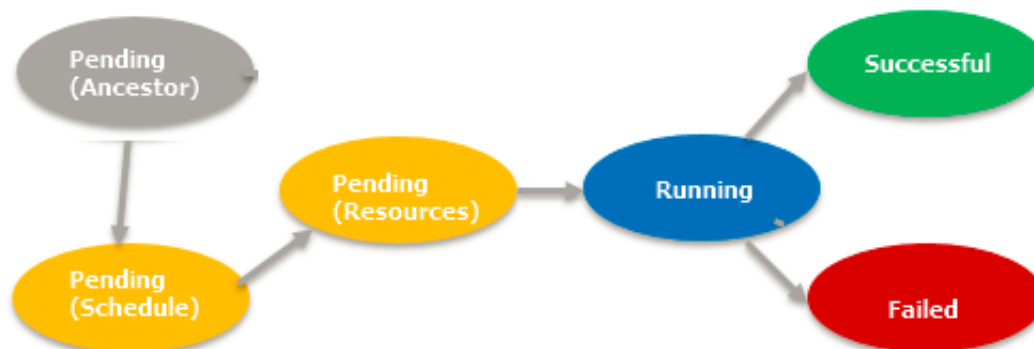
#### 24.4.10.4 State machine

Workflow state machine



- A workflow has four statuses: Not Running, Running, Successful, and Failed.
- The initial status of a workflow is Not Running. At this time, all instances in this workflow are in the Not Running status. When the workflow is invoked by the scheduling system, its status changes to Running and the root instance of the workflow runs.
- When an instance in the workflow fails, the status of the workflow changes to Failed.
- When all instances in the workflow are in the Successful status, the status of the workflow changes to Successful.

Task instance state machine



- A task instance has six statuses: Not Running, Waiting for Scheduled Time, Waiting for Resources, Running, Successful, and Failed.
- The initial status of a task instance is Not Running. When it is invoked by the scheduling system, the system checks whether all its predecessor tasks are in

the Successful status. If yes, the status of the instance changes to Waiting for Resources.

- The task instance is invoked at the time that is specified for running the task. The instance is then sent to the execution engine and its status changes to Waiting for Resources.
- The execution engine allocates resources to the instance. The instance runs, and the scheduling system changes the status of the instance to Running. The execution engine then sends the result to the scheduling system, and then the scheduling system changes the instance status (to Successful or Failed) accordingly.

### 24.4.10.5 Task dependencies

You can configure dependencies for tasks based on your business requirements.

#### Same-period dependency

This is the most common scenario where an instance only depends on its parent instances in the same day. You can configure the following dependencies: A daily instance depends on another daily instance, a daily instance depends on an hourly instance, an hourly instance depends on a daily instance, or an hourly instance depends on another hourly instance.

If an hourly instance depends on another hourly instance, three situations can occur: The number of parent instances is equal to the number of child instances, the number of parent instances is greater than the number of child instances, or the number of parent instances is less than number of child instances. The following examples show all the situations.

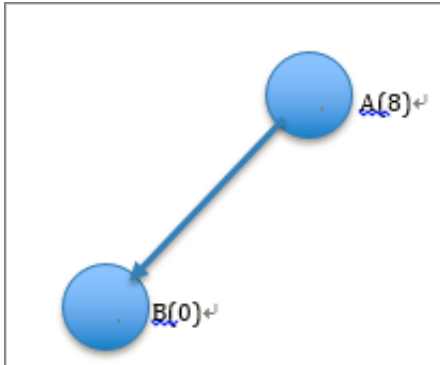


#### Note:

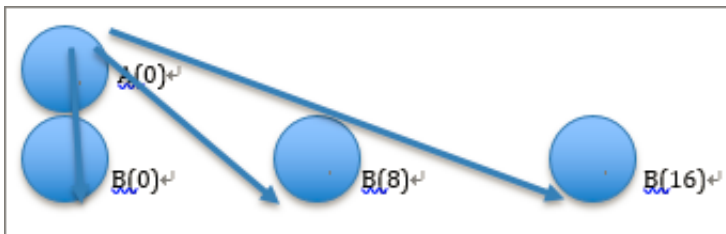
In the following examples, all A nodes are parent nodes, and all B nodes are child nodes.



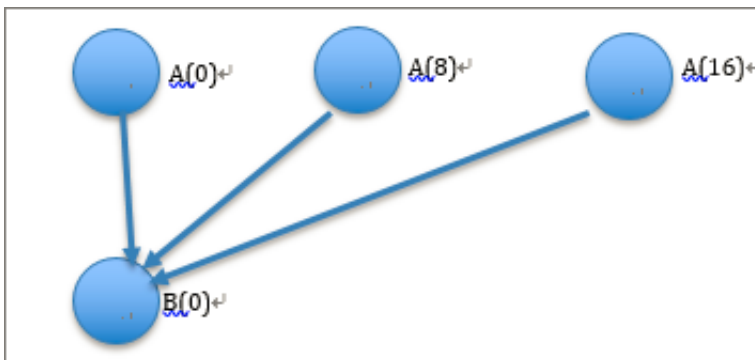
- A daily instance depends on a daily instance. The B node is specified to run at 08:00. The A node is specified to run at 00:00.



- An hourly instance depends on a daily instance. The B node is specified to run at 00:00, 08:00, and 16:00. The A node is specified to run at 00:00.

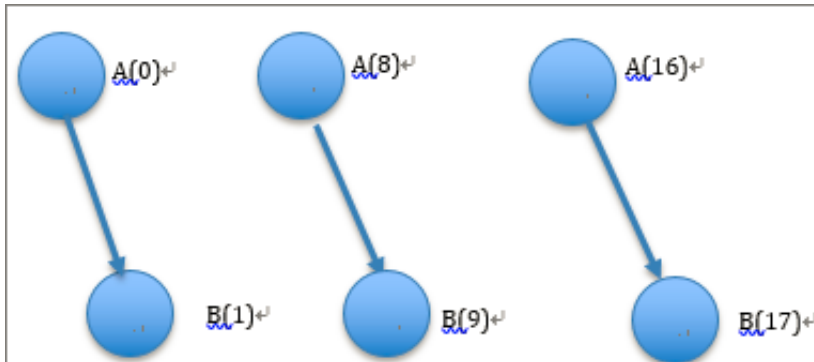


- A daily instance depends on an hourly instance. The B node is specified to run at 00:00. The A node is specified to run at 00:00, 08:00, and 16:00.

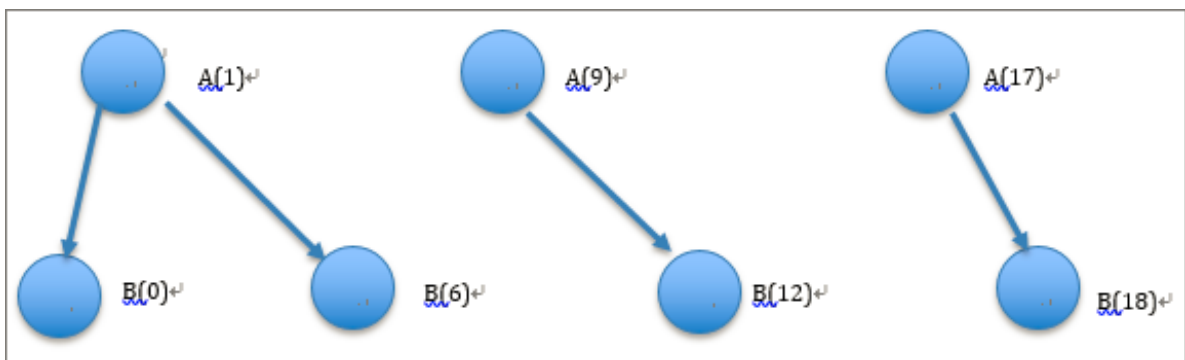


- An hourly instance depends on an hourly instance, and the number of parent instances is equal to the number of child instances. The B node is specified to

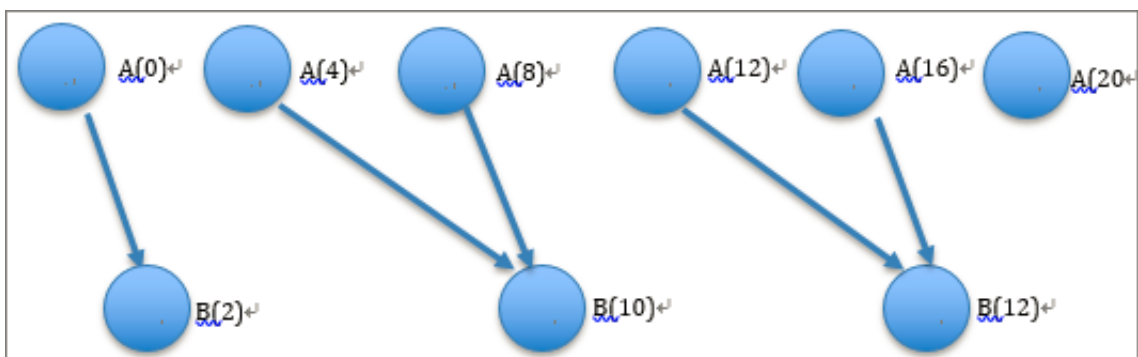
run at 01:00, 09:00, and 17:00. The A node is specified to run at 00:00, 08:00, and 16:00.



- An hourly instance depends on an hourly instance, and the number of parent instances is less than the number of child instances. The B node is specified to run at 00:00, 06:00, 12:00, and 18:00. The A node is specified to run at 01:00, 09:00, and 17:00.



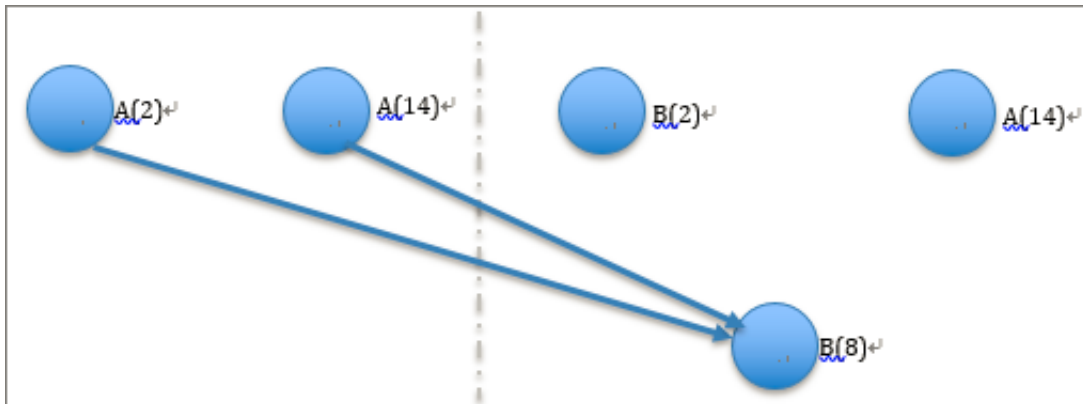
- An hourly instance depends on an hourly instance, and the number of parent instances is greater than the number of child instances. The B node is specified to run at 02:00, 10:00, and 18:00. The A node is specified to run at 00:00, 04:00, 12:00, 16:00 and 20:00.



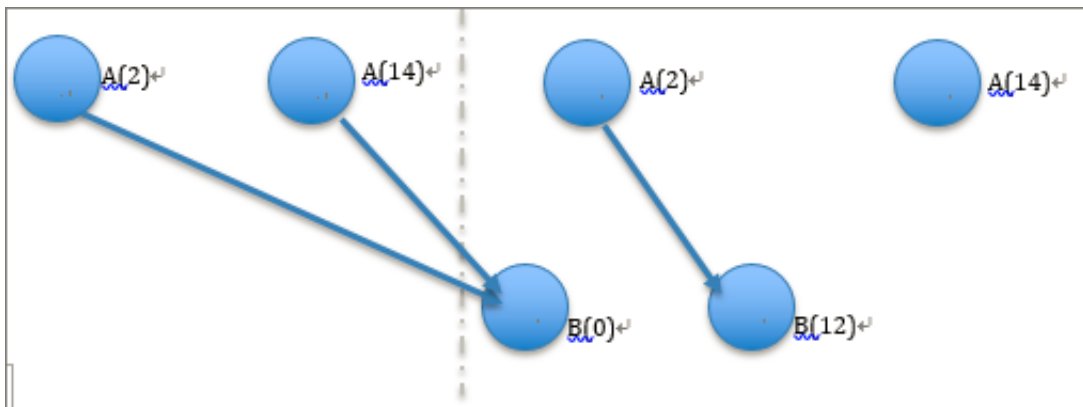
### Cross-period dependency

**You can configure cross-period dependency if the data processing operation requires the result of the data processing operation on the previous day.**

- In most cases, you only need to configure the dependency between the current instance and the instance in the last day. Suppose that the A node is specified to run at 02:00 and 14:00, and the B node is specified to run at 08:00.

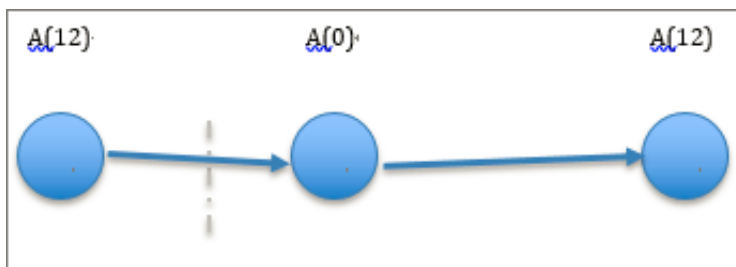


- The same period dependency and the cross period dependency can both exist. Suppose that the A node is specified to run at 02:00 and 14:00, and the B node is specified to run at 00:00 and 12:00.



### Self-dependency

If a task instance depends on the instance that is generated from the same task in the last period, you need to configure self-dependency. The following figure shows the dependencies in the situation where the A node is specified to run at 00:00 and 12:00.



## 25 Realtime Compute

---

### 25.1 What is Realtime Compute?

#### 25.1.1 Background

Realtime Compute has its beginnings in the real-time big screen service of Alibaba Group during the Double 11 Shopping Festival. The big screen service allows you to view sales data during the shopping festival in real time on big screens. With five years of experience and development, the small team that once provided the real-time big screen service and limited real-time reporting services has become an independent and reliable cloud computing team. Realtime Compute provides an end-to-end cloud solution for stream processing based on years of experience in real-time computing products, architecture, and business scenarios. We strive to help more enterprises with real-time big data processing.

We previously used the open source Storm system to support the big screen service of Alibaba Group during the Double 11 Shopping Festival. We also developed stream processing code based on Storm. In these early stages, the stream processing service was provided on a small scale. Developers used Storm APIs to create jobs for stream processing. In this scenario, developers must have proficient technical skills, handle debugging challenges, and perform large amounts of repetitive work.

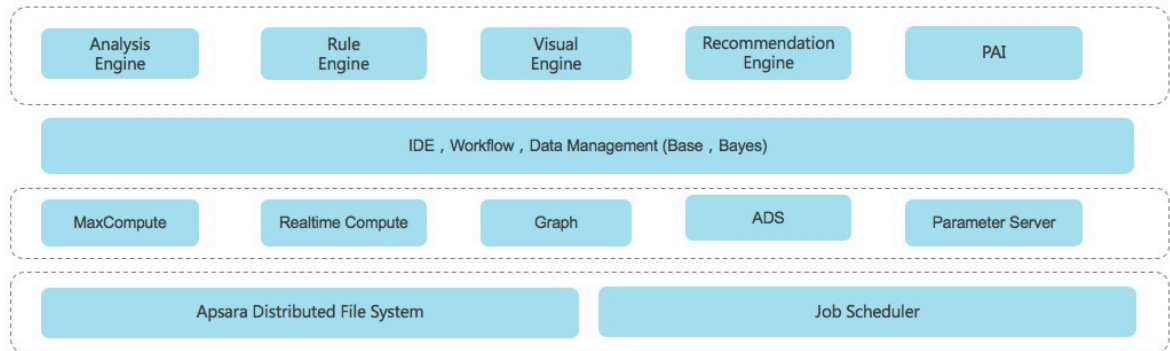
To address these challenges, we started working on data encapsulation and abstraction. Before data encapsulation and abstraction, we needed to choose an integrated processing engine for stream and batch processing from the available options: Apache Spark and Flink. The key difference of Apache Spark and Flink lies in the way they process data streams and batches. In Apache Spark, data streams are divided into micro batches, which are then processed by the Spark engine to generate the final stream of results in batches. For this method, the overhead must be increased to achieve a lower delay. Therefore, it is hard to reduce the delay of Spark Streaming to seconds or to sub-second level. In Apache Flink, batches are considered as bounded data streams that have a defined start and end. In this way, most code can be shared for stream and batch processing, which allows you to leverage the advantages of batch processing. Based on a thorough comparison

between Apache Spark and Flink, we decided to use Apache Flink as the processing engine for real-time computations over data streams. Stream processing methods can be classified as stateful computations and stateless computations. The introduction of state management allows you to easily implement complex processing logic, which is ground-breaking for stream processing.

Any emerging technology is only adopted by a small group in the beginning. With the growth of this technology and the reduction in adoption costs, it will be widely accepted. Therefore, we are working to enable stream processing technologies to be widely adopted by improving the technology and decreasing adoption costs. Apache Flink has made many improvements to the architecture, but its implementation mechanism needs to be optimized. For example, the tasks of multiple jobs may be executed by the same thread, which greatly reduces the computing performance. To resolve this issue, we introduce the YARN system. Another example is the checkpoint feature of Apache Flink. In Apache Flink, checkpoints are created to ensure data consistency, but checkpoints cannot be created when the state stored for incremental computing is excessively large. To address this challenge, Realtime Compute optimizes the checkpoint feature to efficiently manage large state. Realtime Compute has addressed many performance issues and bottlenecks to ensure the stability and scalability in the production environment. Currently, Realtime Compute is capable of supporting core businesses. We have also improved the SQL of Realtime Compute to support complex business scenarios. We are working to provide excellent user experience through constant exploration and innovation.

### 25.1.2 Key challenges of Realtime Compute

Realtime Compute runs on a cluster of thousands of nodes within Alibaba Group. It provides services for hundreds of real-time applications for over 20 business units of Alibaba Group, processing hundreds of billions of messages and about 1 petabyte of traffic per day. Realtime Compute has become one of the core distributed computing services of Alibaba Group.



We are working to make the following improvements:

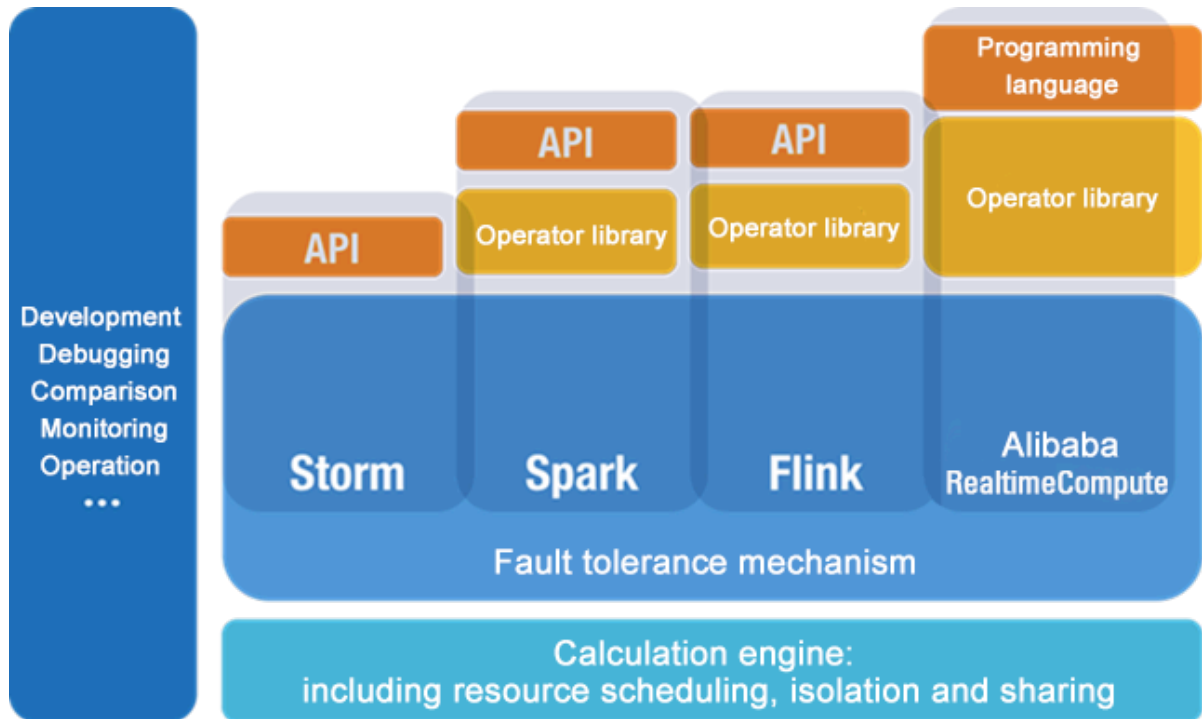
- **Computing engine:** We are working to improve the engine performance and enable the engine to support multiple semantics of processing messages.
- **Programming interfaces:** We are working to enable support for more APIs and programming languages. For example, we are working on the compatibility with open source APIs, such as Storm APIs and Beam APIs.
- **Programming languages:** We are working to enable support for more SQL syntaxes and semantics in stream analysis scenarios, such as temporal tables and complex event processing (CEP).
- **Services:** We are working to improve Realtime Compute from the following aspects: debugging, one-click deployment, hot upgrades, and training systems.

## 25.2 Technical advantages

Realtime Compute uses a compute engine that is developed based on Apache Flink, which allows Realtime Compute to leverage advantages of Apache Flink and optimize the Flink Table API. You can use Flink SQL for batch and stream

processing. The application of YARN in Realtime Compute enables full compatibility with Flink API, which enables a large ecosystem of stream processing.

Figure 25-1: Realtime Compute and other stream processing system



*Figure 25-1: Realtime Compute and other stream processing system* shows the differences between the technologies of Realtime Compute and other stream processing systems. Based on the extensive experience of addressing challenging business scenarios, Realtime Compute provides the following benefits:

- **Powerful stream processing functions**

Unlike these open source systems, Realtime Compute simplifies the development process by integrating a wide range of functions. These functions are described as follows:

- A powerful engine is used. This engine offers the following advantages:
  - Provides the standard Flink SQL that enables automatic data recovery from failures. This ensures accurate data processing when failures occur.
  - Supports multiple types of built-in functions, such as text functions, date and time functions, and statistics functions.
  - Enables an accurate control over computing resources. This ensures complete isolation of each tenant's jobs.
- The key performance metrics of Realtime Compute are three to four times higher than those of Apache Flink. For example, in Realtime Compute, the data processing delay is reduced to seconds or even to sub-second level. The throughput of a job reaches millions of data records per second. A cluster can contain thousands of nodes.
- Realtime Compute integrates cloud-based data stores such as MaxCompute, DataHub, Log Service, ApsaraDB for RDS, and Table Store. With Realtime Compute, you 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 user-developed stream processing services, Realtime Compute is a fully managed stream processing engine. You can query streaming data without deploying or managing any infrastructure. With Realtime Compute, you can use streaming data processing services with a few clicks. Realtime Compute integrates services such as development, 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. This isolation and protection extends from the top application layer to the underlying infrastructure layer. This helps to ensure the security and privacy of your data.



- **Excellent user experience during development**

**Realtime Compute provides a standard SQL engine: Flink SQL. It also provides many built-in functions, such as the text functions, date and time functions, and statistics functions. The application of these functions greatly simplifies and accelerates the Flink-based development. With Flink SQL, even users with limited development knowledge, such as business intelligence (BI) analysts and marketers, can easily perform real-time analysis and processing of big data.**

**Realtime Compute provides an end-to-end solution for stream processing, including development, administration, monitoring, and alerting. On the Realtime Compute development platform, only three steps are required to publish a job.**

- **Low costs**

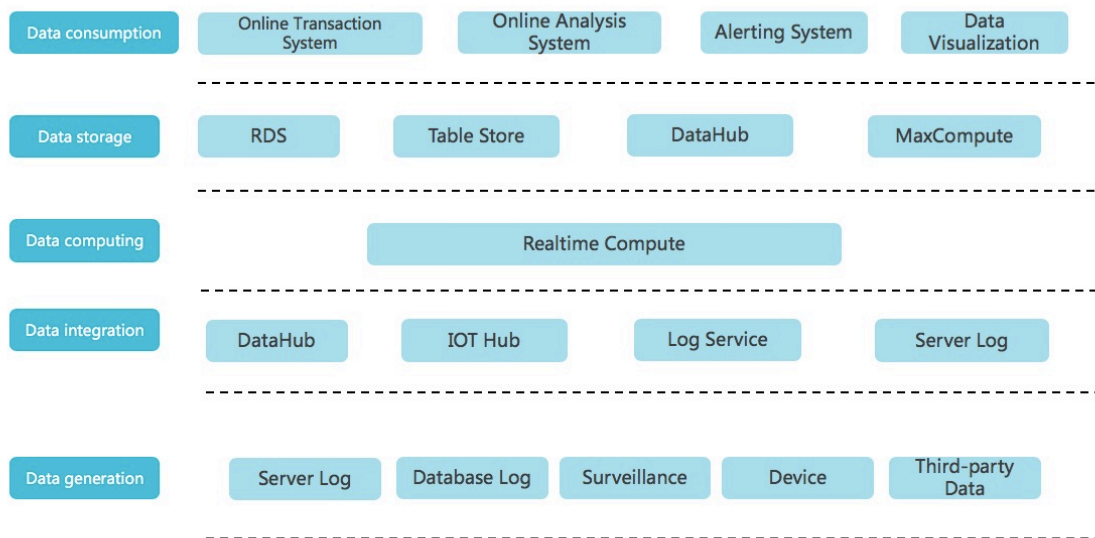
**We have made many improvements to the SQL execution engine, allowing you to create jobs more cost-effectively than to create Flink jobs. Realtime Compute is more cost-effective than open source stream frameworks in both development and production costs.**

## 25.3 Product architecture

### 25.3.1 Business architecture

**Realtime Compute is a lightweight SQL-enabled streaming engine for real-time processing and analysis of data streams.**

Figure 25-2: 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 storage

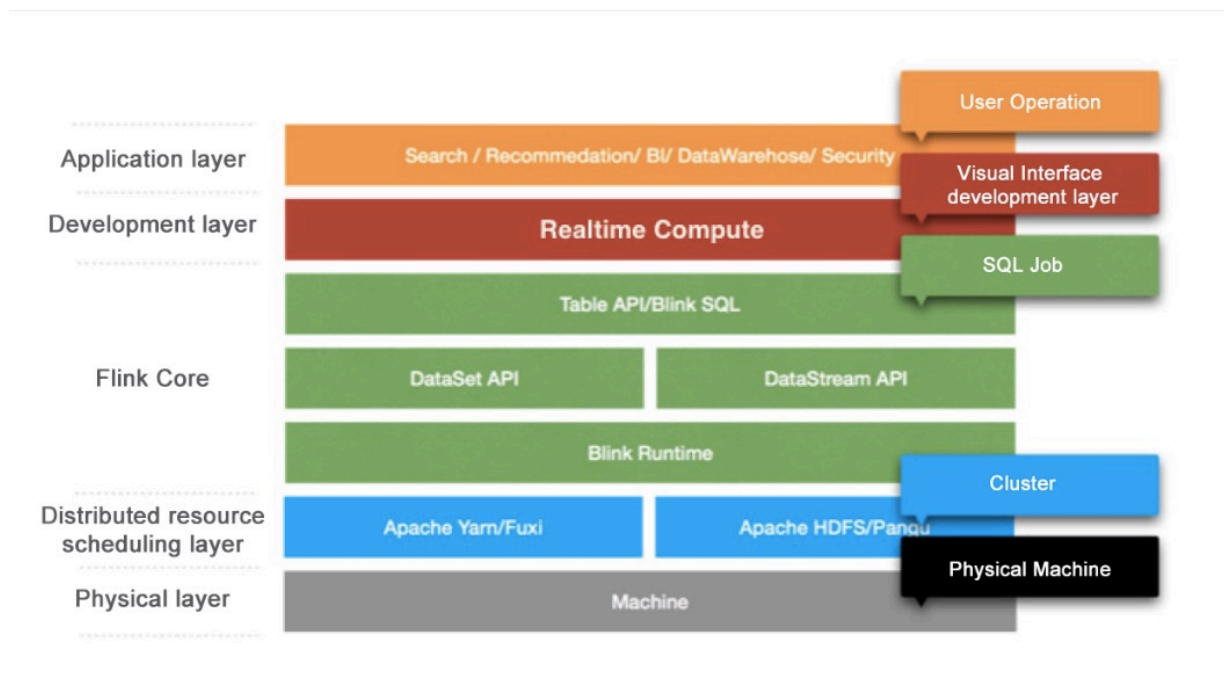
Realtime Compute does not provide built-in data stores. Instead, it writes computing results to external data stores, such as relational databases, NoSQL databases, and online analytical processing (OLAP) systems.

- 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.

### 25.3.2 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.

## 25.4 Functional principles

**The Blink engine of Realtime Compute is developed based on Apache Flink. For more information about the functional principles of Realtime Compute, see**

*[Discussion on Apache Flink](#).*

## 26 DataQ - Smart Tag Service

---

### 26.1 What is DataQ - Smart Tag Service?

#### 26.1.1 Overview

**DataQ - Smart Tag Service is a tag-oriented service, which establishes a unified logic model across multiple schemas. By using the tag model view, developers can integrate the data service modules with profile analysis, rule warnings, text mining, personalized recommendations, relational networks, and other business scenarios. This helps developers use APIs to quickly build applications.**

**The IT team can share tags that are frequently used in a diverse range of business scenarios. You can apply for using these shared tags. After you are authorized to use these tags, you can perform corresponding computations by calling APIs. You can also generate code that can be independently deployed by configuring parameters in the console. This helps provide an easy way to build the corresponding big data product.**

#### Components

- **Tag center**
  - **Tag models**
  - **Tag warehouse**
  - **My tags**
  - **The overview chart**
  - **Model views**
  - **Schemas**
  - **Data import**
- **Analysis APIs**
  - **APIs**
  - **API factory**
  - **Fast search**

- **Dashboards**
  - **Datasets**
  - **Report configurations**
  - **Report permissions**
- **Tag factory**
  - **Tag schemes**
  - **Tag tasks**
- **Tag sync**
  - **Sync schedules**
  - **Sync tasks**
  - **Task O&M**
- **Homepage**
  - **Overview**
  - **Core data assets**
  - **Tag statistics**

### 26.1.2 Current situation

With the rapid development of Internet and big data technologies in recent years, various data products have emerged. The development of Alibaba big data applications is fast, but still faces the following challenges:

- To cope with the rapid growth of data volumes, various types of distributed data computing and storage technologies are developed to solve many difficulties across diverse application scenarios. In a non-traditional IT architecture, only a single database is required to support data analysis reports for the entire enterprise. The methods for integrating and managing various types of data, merging various business databases, and managing the distribution of multiple computing and storage resources has become a major challenge.
- Big data is used in various industries, such as digital advertising, Internet finance, e-commerce, and online security and risk control. A data application includes report analysis, behavior prediction, real-time monitoring, credit scoring, personalized recommendations, text mining, and spatiotemporal data. It integrates various big data technologies rather than only generating report statistics for enterprise operations.

- Currently, the target data users are not limited to professional data analysts and data warehouse engineers, but also include the business personnel who have limited technical knowledge. This requires a system to help them perform data exploration in an easy and cost-effective way.
- Therefore, if you want to make good use of big data, you must have the ability to design an enterprise IT infrastructure to meet your complex and diverse business needs. The technical engineers must also have the following comprehensive abilities and skills:
  - Understand the characteristics of various types of distributed computing and storage resources.
  - Have an in-depth understanding of the data usage scenarios of business personnel and then help develop business-oriented data products.
  - Design capable architectures for these computing and storage resources for various application scenarios, such as data analysis and algorithm services.

### 26.1.3 Scenarios

In addition to configuring modules in the console, you can also integrate operations related to data elements and data services of modules to your own application systems by using APIs. This delivery mode facilitates system integration and data application management.

From the perspective of IT architecture, IT or data departments can deliver the data service to business departments and partners for development by using DataQ - Smart Tag Service. The data service is packaged with computing resources, data resources, and algorithms.

This delivery mode is convenient for application developers and allows them to activate and use DataQ - Smart Tag Service. Furthermore, this delivery mode adds further convenience for IT departments to effectively manage platform resources and reduce redundant data storage and processing. This is particularly valuable for business algorithms and consumer profiles that require detailed data computing. The following benefits are provided:

- Detailed data can be fully leveraged.
- The production of raw data is not affected and no redundant copies of large data volumes are created.



- This reduces the threshold for data usage and provides powerful assistance and support.

## 26.1.4 Product benefits

Alibaba Cloud DataQ - Smart Tag Service provides a data IDE to accelerate the development and implementation of big data applications. This product helps developers integrate various big data products based on their business needs, which reduces most of the engineering workload that is necessary for building big data applications. By using the product together with relevant industry application solutions, developers who are less experienced in the development of big data applications can quickly build big data applications. This can help realize the true value of big data over a relatively short period of time.

DataQ - Smart Tag Service provides the following benefits:

- Simplifies the integration with complex systems because application developers do not require a deep understanding of multiple underlying computing and storage resources.
- Helps the IT team to manage data usage by providing data service APIs. This helps to avoid any duplication and redundancy of resources.

## 26.2 Technical benefits

Modeling across computing resources (schemas)

In traditional modeling, the data sources that are used for modeling are derived from the same database. In the big data environment, data sources used for modeling are distributed across multiple computing and storage resources because of various data acquisition methods and diverse data computations. Data generation and processing may be across multiple databases. Data must be calculated across streaming, ad hoc multidimensional analysis, offline algorithm for processing, and other methods. This requires data transmission across multiple storage and computing resources.

Management from the business perspective

Tag modeling is based on three elements: objects, links, and tags (OLT). Data in the OLT model is organized and managed from the business perspective, rather than modeling based on the table concept. The OLT model is similar to a conceptual

**model and is easier to be understood. This helps you to understand and manage data at the application layer.**

#### Flexible scalability

**You can manage the model in a dynamic manner because the relationship between tables and tags is established at the logic layer. This simplifies the model management and maintenance without consolidating data at the physical layer. Each tag can be used independently. The data can be used more flexibly by using tags that are attached to discrete columns across multiple schemas.**

#### Accelerating application development

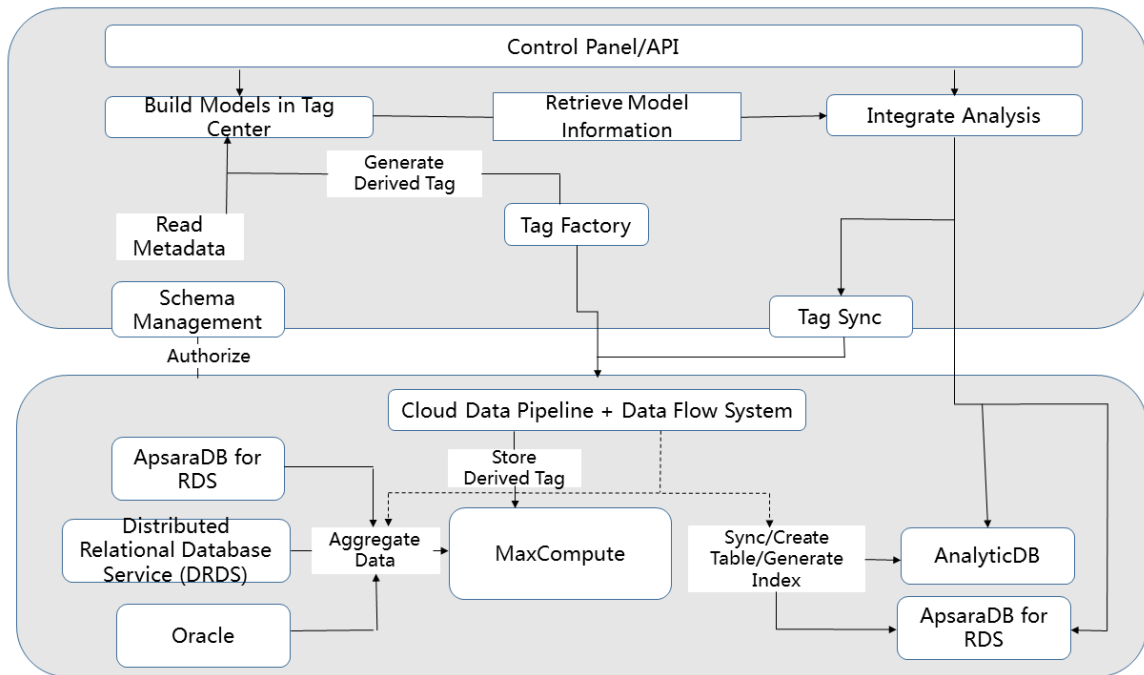
**DataQ - Smart Tag Service integrates computing resources, data resources, and algorithms into a unified package as a data service. This service package can be directly provided to business departments and partners for easy use.**

## 26.3 Production architecture

**In big data environments, a single data application often relies on multiple computing resources. In general, data first needs to be processed offline and then synchronized to online databases for analyzing and querying. During the process, the tag center obtains the data elements of computing and storage resources by communicating with multiple databases. These data elements are used for logical modeling. The tag center also parses the instructions that are sent from various data service modules into calculation commands and sends these commands to each computing resource.**

**The following section uses the analysis APIs as an example to describe the overall architecture of DataQ - Smart Tag Service.**

**In the most common OLAP analysis scenario, data is extracted from a service database and loaded to MaxCompute to be processed and derived. Then, the data to be analyzed is synchronized to online analytical databases, such as AnalyticDB that is used for analyzing a large amount of data.**



You can authorize DataQ - Smart Tag Service to access your schemas by calling APIs or using the DataQ - Smart Tag Service console. After the authorization, DataQ - Smart Tag Service can read the data elements of your schemas. After modeling and configuration, you can manually or automatically trigger the tag sync function and send associated data sync tasks to DataWorks and Cloud Data Pipeline (CDP). In this way, you can integrate a large amount of data at the granularity of tag from service databases to offline data warehouses. You can also synchronize data from service databases to online analytical databases, and create tables and indexes.

Then, you can perform calculations based on the tag model views in the console or by using the APIs for data service. For those parts that need to be calculated offline, some common operations can be processed by creating derived tags in the tag factory for integration into MaxCompute. For example, you can create derived tags by aggregation or filtering conditions.

During the entire process, DataQ - Smart Tag Service integrates different architectures across multiple schemas to satisfy common business scenarios. This simplifies the system integration process on the big data platform.

## 26.4 Features

### 26.4.1 Tag center

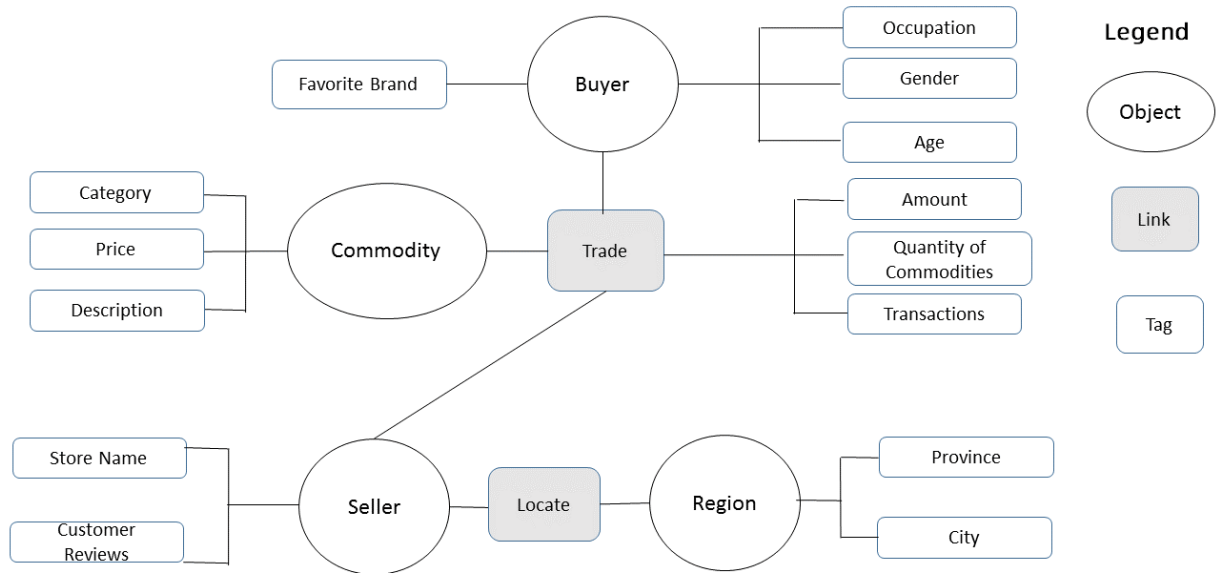
#### 26.4.1.1 Overview

The tag center is used to build a logic model across computing and storage resources based on existing data tables. This allows you to manage, process, and query data at the tag model layer without interacting with underlying big data computing and storage resources. The tag center plays an important role when the data architecture is complex and the combination of multiple computing and storage resources is required.

The tag modeling method is widely used in precision marketing, personalized recommendation, user profiling, credit scoring, and other big data applications based on detailed data computing. A tag is the minimum unit of description for a user object and represents an abstract expression of a specific descriptive fact of an object. The abstract expression, such as attributes, behaviors, and interests, is a data modeling method from the business perspective. For example, attributes include gender (the tag value is male or female) and age (the tag value is the actual age). Behaviors include turnover, bookmarks, and location. Interests include preference for multiple keywords. A tag can be a column consisting of values, enumerated values, and multiple key values, or a fact table consisting of multiple fields (subjects, time, predicates, and objects). In terms of conceptual model, the tag system is a tag-based description methodology. This is built around multiple objects (buyer, seller, commodity, enterprise, and equipment) and the links between objects (transactions).

*Figure 26-1: Tag modeling* shows the tag modeling process.

Figure 26-1: Tag modeling



This modeling method appears similar to anchor modeling or graph data modeling, while actually it is quite different. In traditional modeling, the concept and logic model are first designed according to business needs, and then the physical data tables are processed and sorted based on the logic model. In tag modeling, the logic model is directly built based on existing physical data or models. With the parsing of different data service agents, you can perform various computations on the model view without preprocessing a large amount of physical data.

Tags are created based on the data of physical tables. In a cross-computing context, you may experience differences in query languages and performance between multiple computations. Therefore, tags created on logical requests may not be computed. Each tag you have defined still needs to be associated with the corresponding physical table. In DataQ - Smart Tag Service, you can define the computing logic of a query as a temporary tag in the corresponding data service. However, when the computing logic is related to cross-computing, it needs to be converted into the data of physical tables to avoid any errors.

## 26.4.1.2 Scenarios

The tag center is a cross-computing storage that supports logical and dynamic modeling based on physical models (object-link-tag model). It integrates with data services to provide data modeling and data management tools for big data application and development. The tag center can clearly display the data model view of an enterprise through visual methods.

The tag center is applicable to the following scenarios:

- **Business-oriented data modeling and management**

The tag center provides a tool for data discovery and model exploration from the business perspective. It offers ease and extra convenience for business personnel, developers, and database administrators to gain a deeper insight into enterprise data assets.

- **Centralized business view for diverse data services**

Provides a centralized data view for business data across multiple computing engines, and integrates data services for easier business logic computing.

- **Data permission management**

You can control data access permissions at the logic layer. The permission control is precise and accurate to column level.

## 26.4.1.3 Components

### 26.4.1.3.1 Tag models

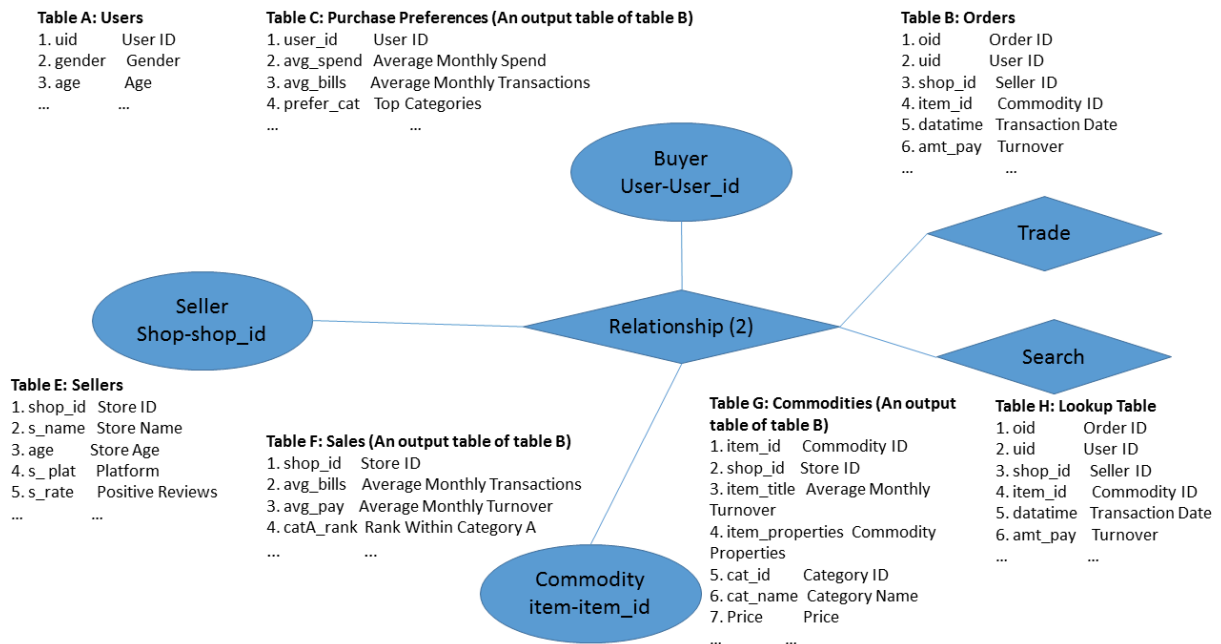
Tag modeling is a network-based modeling method for data distributed in different databases based on three elements of an OLT model: object, link, and tag.

An object is used to describe a real object such as device, personnel, and address, which corresponds to physical data tables (usually property tables). In this table, the primary key represents the object and other columns are tags (namely, the properties of the described object).

Links are relationships, events, and actions between objects. They correspond to physical tables, which are usually fact tables. For example, a deal, repair, and ride.

A tag is attached to an object or link to describe attributes. For example, the device production time and the device usage frequency are tags of the device. The repair time and number of repairs are the tags of a repair.

Compared with the metric-dimension system, this modeling method is more suitable for the description and expression of detailed data. Detailed data mainly consists of a fact table. The concept of links corresponding to the fact table is introduced to show a clear representation of the relationships between multiple objects. This concept is conducive to management and expression and serves as an important step during the analysis phase. On the business side, it is similar to the conceptual model design and is easier to be understood.



After modeling and conversion, you can transform the model into the preceding table to the logical relationship shown in the preceding figure. Transaction tables are mapped to the links, while the amount and time are the tags of the links. The user tables and commodity tables are mapped to buyer and commodity respectively, while gender and age are the tags of the buyer. This modeling method is applicable to scenarios where analyses are performed based on detailed behaviors and relational data.

The Tag Center page consists of seven modules: tag warehouse, my tags, tag models , overview chart, model views, schemas, and data import.

Object-link model management is the primary function of tag center that is used to configure the logic model. It can read metadata from different database sources and integrate the metadata as an object or link. Multiple tables describing the same object (primary key) can be accumulated into a large wide table at the logic layer. The composite primary key tables can be considered as links during creation and

used to associate multiple objects. Other descriptive fields are defined by using tags as required.

#### 26.4.1.3.2 Tag warehouse

Tag warehouse stores shared tags. You can view and apply for shared tags in tag warehouse.

On the Tag Warehouse page, you can apply to use the shared tags that are in the upper-level workspace of the current workspace. You can also quickly search for all shared tags in the corresponding workspaces. Managing shared tags in the workspace based on different categories is also supported.

#### 26.4.1.3.3 My tags

The My Tags page displays private tags, claimed tags, and shared tags.

As a department member, you can view, search, modify, and share private tags. You can also perform fast search, share multiple tags, revoke tag sharing, and detach private tags.

Shared tags are the tags shared by my workspace and sub workspaces to the tag warehouse. These tags are also authorized tags that are available for other users to use. When you want to use the shared tags in the tag warehouse, you need to click Apply to submit an application and the workspace administrator needs to approve the application.

#### 26.4.1.3.4 The overview chart

You can view and analyze the entire tag model through the overview chart.

On the Overview page, you can view the information of all objects, the relationships, and attributes between these objects, and tags attached to objects by using an entity relationship (ER) diagram. The information includes the name, code, descriptions, creation time, associated tables, and attached tags of an object or link. You can also create tags for objects or links, and view the list of associated tables and tags.

#### 26.4.1.3.5 Model views

When the number of business models is large, the relationships between business models are difficult to be analyzed in the overview chart. On the Model View page, you can drag and drop an object or a link from the search entities to create an



intuitive model view. This model view is a sub chart of the overview chart and allows you to quickly find the required data in a complex model.

#### 26.4.1.3.6 Schemas

Schema management supports communications between multiple computing and storage resources to obtain metadata.

DataQ - Smart Tag Service allows you to manage the following computing and storage resources.

- ApsaraDB for RDS
- MaxCompute
- AnalyticDB (ADS)
- Table Store
- DataHub
- Realtime Compute

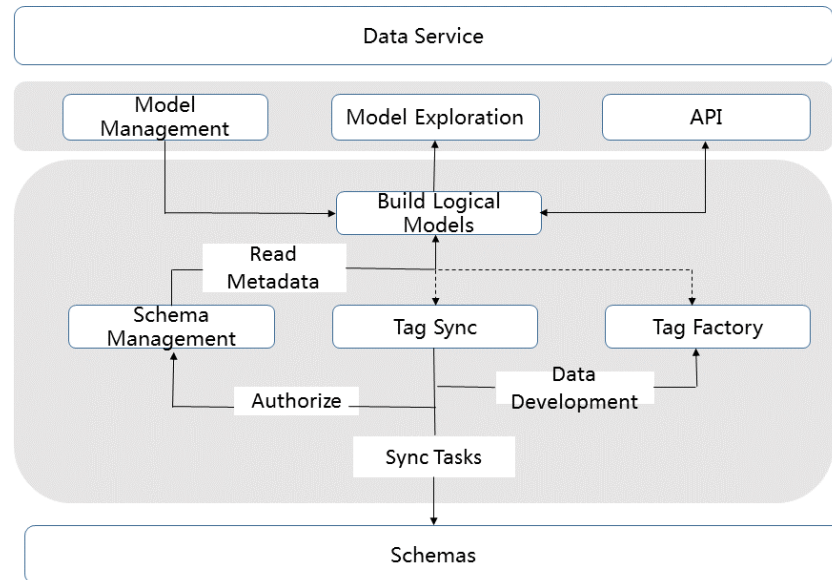
#### 26.4.1.3.7 Data import

The tag center supports the following two types of uploading files: TXT and CSV . You can import tasks into schemas (only MaxCompute schema is currently supported) to create tags.

### 26.4.1.4 Technical architecture

*Figure 26-2: Technical architecture* shows the technical architecture of Tag Center.

Figure 26-2: Technical architecture



### 26.4.1.5 Features

Data system planning is driven by business requirements and data accumulation. With the development of business units, more and more data sources are involved in the design of system. When you use the traditional data warehouse to process data, the following issues may occur:

- Data tables need to be frequently consolidated at the physical layer. The frequent changes of underlying data tables may cause instability in data usage.
- More and more tags are involved in the system. It is difficult to use a single raw table at the physical layer to consolidate data without limits. The more scattered are the tables, the more tags that are attached to these tables. Therefore, managing these tags and tables becomes a quite significant challenge.
- It is difficult to use a large table for various applications. Instead, you must use several columns across multiple tables. The continuous extraction and integration of multiple applications will lead to difficulties in management and retrieval.
- Tags can either be real-time data or offline data. Different data storage methods make the tags difficult to manage.

Compared with traditional BI modeling, tag modeling has the following benefits.

- **Management from the business perspective**

Tag modeling is based on three elements: objects, links, and tags (OLT). Data in the OLT model is organized and managed from the business perspective, rather than modeling that is strictly based on traditional table models. The OLT model is similar to a conceptual model and is easier to be understood. This helps you to understand and manage data at the application layer.

- **Centralized logic model across computing resources (schemas)**

In traditional modeling, the data sources that are used for modeling are derived from the same database. In the big data environment, data sources used for modeling are distributed across multiple computing and storage resources because of various data acquisition methods and diverse data computations. Data generation and processing may take place across multiple databases. Data must be calculated across streaming, ad hoc multidimensional analysis, offline algorithm for processing, and other methods. This requires heavy data transmission across multiple storage and computing resources.

The tag system establishes unique field mappings between multiple schemas and a logical view. To be specific, a tag of the logical view is mapped to physical table fields across multiple schemas.

- **Flexible scalability**

You can manage the model in a dynamic manner because the relationship between tables and tags is established at the logic layer. This simplifies the model operations and maintenance without consolidating data at the physical layer. Each tag can be used independently. The data can be used more flexibly by using tags that are attached to discrete columns across multiple schemas.

With the improvement of computing capability and diversity of application scenarios, data computing tends to calculate detailed data of user behaviors rather than perform multi-dimensional statistics of metrics. The metrics and multi-dimensional statistics architecture in traditional data warehouse modeling are only applicable to a few business scenarios. Tags can be defined as data of various data types to achieve flexible operations. The value of a tag can be either a single column data or a combination of dimensions and tags. Note: The combination of dimensions and tags is usually used to describe a user behavior.

## 26.4.2 Analysis APIs

### 26.4.2.1 Overview

**This module is built on a tag-based view to provide business functions. You can use tags as dimensions to perform unified computations for data across multiple computing resources by configuring APIs or calling API operations.**

**The combination of data service and dynamic logic modeling reduces the workload and offers high scalability. Especially when the big data environment needs to consolidate data from multiple systems, it is difficult to design a single plan to meet all data requirements.**

**From the perspective of applications, the tag model allows you to calculate and query detailed data by using an intuitive tag system without the need for a complex data structure. This also helps you to optimize the process of data development and application development.**

### 26.4.2.2 Scenarios

**The Analysis APIs module can be used in conjunction with AnalyticDB to integrate data distributed across multiple storage resources. This helps you to build an interactive big data profile analysis application based on the tag model. Therefore, your business personnel can freely and flexibly analyze the correlations between various attributes of these objects and behaviors. It is applicable to profile analysis in various scenarios, such as industrial devices, business operations, and user behaviors.**

**The features of big data profile analysis are described as follows.**

- **Analysis based on behaviors and other detailed data**

**In the past, as analysis was aimed at calculating KPIs, index calculations tended to be hastily built. With the variety of data acquisition and usage scenarios, business personnel want to be able to freely analyze the detailed data of diverse behaviors. The data includes consumption preferences and correlation purchases of different customers under various commodity categories. The data also includes the failure rates and maintenance of devices for different types and attributes at different regions, and specific customers and device lists in different dimensions. The content that is analyzed by business personnel may**

have cross-relationships between dimensions. Therefore, calculating such a diverse plethora of information in advance is a very difficult task.

- **Extracting features from semi-structured data**

Flexible analysis also requires the integration with leading edge practices that include prediction, scoring, text feature extraction, and other algorithm-based technologies. This helps you to perform extensive and in-depth analyses. Many profile features, such as preferences or interests are often extracted from user behaviors by using algorithms. For example, the interests of animations and the preferences of cosmetic are extracted from user clicks, favorites, purchases, and descriptions of related commodities. Therefore, an algorithm that supports preference calculation and text mining is required. This algorithm can help you to perform an in-depth analysis and extract user features from the semi-structured data.

- **Interactive search and analysis**

Analysis is often used to explore and exploit useful information, such as potential factors that affect customer purchases or factors associated with faulty devices. During the analysis process, you need to adjust filtering conditions, dimension set, and drill-down aggregation until results are returned as expect. This process requires a quick response during querying.

Such interactive analysis scenarios also require an organized user experience (UX) design. The business personnel only focus on data insights that are gained through continuous data exploration. However, the data exploration process is affected by the complicated configuration of reports and the knowledge level of technologies. Data analysis also leverages various visual tools such as charts to provide an institute data view. These charts include common diagrams, maps with different granularity (especially the city dimension), network diagrams showing topological relationships, and other diagrams that display useful information such as text features.

Based on the above features of big data profile analysis, the development of interactive data analysis products has become very challenging. The challenges are as follows:

- **You must fully understand the data structure to design a fully comprehensive data architecture. The architecture must provide better cross-table query logic and user interface (UI) interaction.**

- You need to understand the features of various storage and computing resources to design a page that displays a complete overview of results.
- You need to be familiar with the use of various charts and analysis controls and apply them for different types of analysis.

### 26.4.2.3 Components

Analysis APIs consists of the following pages: API Factory, APIs, and Fast Search.

#### API factory

On the API Factory page, you can automatically synchronize data across multiple data sources from tags to tables or indexes of AnalyticDB and relational databases. You can also debug analysis APIs on the page. The entire analysis API is expressed by querying the TQL created on the tag model layer. Attributes related to the same object you are querying can be considered as a wide table. However, the data may be distributed across multiple physical tables.

The API factory allows you to debug analytic statements and encapsulate data analysis APIs. The query expressions of analysis APIs are built on an object-link model.

By debugging the API, you can view query results, runtime errors, time spent parsing syntax for each step, and parsed SQL statements.

#### APIs

You can directly generate APIs for analysis query. Application developers can analyze and query data through APIs. You can also manage API categories, debug APIs, and publish APIs in the Analysis APIs module.

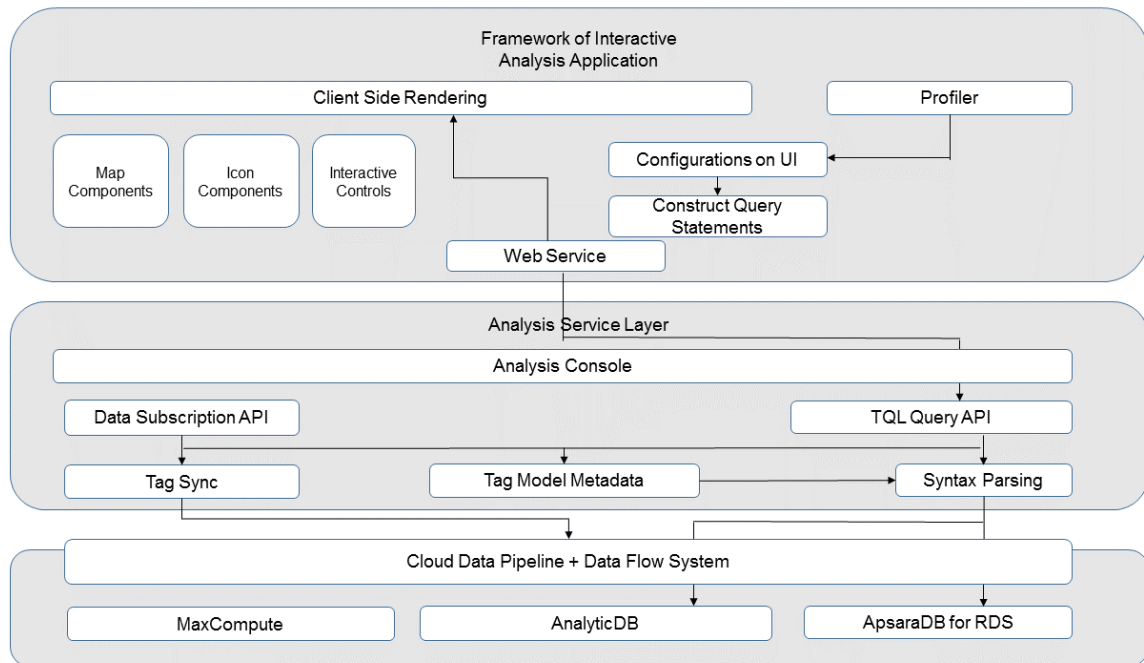
#### Fast search

Filters tags of objects or links across multiple schemas and saves the filtering conditions as APIs. These APIs can be called by other operations or used to generate tag datasets. Datasets can be exported to Excel or generated charts for you to quickly view and understand the characteristics of objects or links. Generated charts can be exported.

## 26.4.2.4 Technical architecture

*Figure 26-3: Technical architecture* shows the technical architecture of Analysis APIs.

Figure 26-3: Technical architecture



The technical architecture of analysis APIs consists of two parts. For the service layer part, input parameters are received by calling API through TQL query, and the SQL syntax is parsed by referring to the data elements in the tag model. Then, the parsed SQL statements are sent to the corresponding schema, and the calculation result is returned through the API. The Debug mode returns both the parsed SQL statements and the elapsed time of each calculation step, which can be used to optimize corresponding query statements. The syntax parsing refers to the process of translating user query logics on the model view to actual JOIN queries among tables.

For the data subscription part, one-click data migration can be completed by using the console. The console invokes the data subscription API to obtain the data elements of corresponding data and then invokes the underlying API of the tag center for tag sync. After a schema creates tables and indexes, a sync task is triggered to synchronize data.

At the application framework layer of interactive analysis, source code of the development and configuration framework is provided. The framework includes the corresponding frontend components, backend web services, configuration files

, and query APIs configured according to the configuration file. Meanwhile, you can configure routing for the configuration file to make sure that different URLs route to correct configurations and different users can view different APIs.

### 26.4.2.5 Features

Analysis APIs provides the following features:

- **One-click data integration**

You can synchronize multiple tags across various schemas to online analytic databases by one click for different analyzed objects. You can also set indexing for these objects and manage schemas in the same way of managing a table. The compatible online analytic databases include AnalyticDB and ApsaraDB for RDS.

- **User-friendly web development**

Web App developers can quickly create their own analytic data products by calling APIs for analysis and tag metadata query and integrating with other Alibaba Cloud visualized products (such as DataV).

- **Simple query expressions**

The tag system effectively simplifies the expression of table joining and subqueries, allowing Web application developers to focus on the logic of applications rather than the logic of table structure. The query parameters can be provided in JSON format or in the TQL statements that are similar to SQL statements.

- **Seamless integration with other modules of DataQ - Smart Tag Service**

Based on the tag system architecture, multiple modules share the same tag view . The same tag can be automatically synchronized between different storage and computing resources. This guarantees the consistent expressions of data generated by Analysis APIs, algorithms, feature engineering, and real-time monitoring and alerts to achieve the seamless integration of these modules.

- **Interactive analytical application framework**

Provides a tool by using SDK to configure analysis APIs, and immediately generates an interactive analysis application. Compared with traditional BI tools , the analysis API is similar to an independent and interactive analysis product , which can be integrated into your entire analysis system. It is easier for you to accurately analyze entity attributes, behaviors, and locations.



## 26.4.3 Tag factory

### 26.4.3.1 Overview

The Tag Factory module provides the processing of business tags, including the processing of tag schemes and tag tasks.

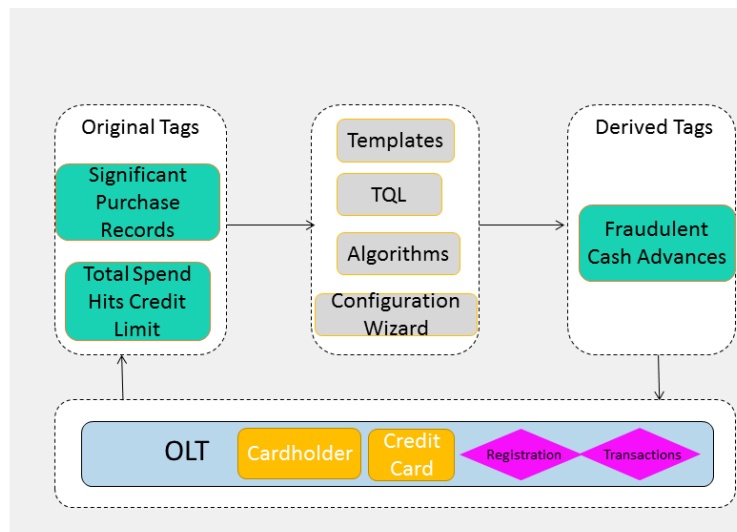
On the Tag Schemes page, you can create tag schemes by configuring TQL and algorithms. On the Tag Tasks page, you can run the tag tasks that are generated by the configured tag schemes.

### 26.4.3.2 Scenarios

The tag factory is applicable to scenarios involving the requirements of flexible tag generation and insufficient human resources of data processing. This reduces the requirements for developing derived tags and allows business personnel who have the ability of configuring simple TQL statements to configure derived tags as expected.

Based on existing tags, tag construction is used to carry out feature engineering of derived classes for common tags, or to extract structured tags from unstructured text data.

Feature engineering is an assistant tool to develop derived methods for existing tags. For example, when analyzing the consumption behaviors of an individual, more information will be calculated based on the original transaction details of the individual. This information includes the monthly average consumption amount, category preferences, and purchase frequency. Alternatively, the combination of conditions that are used frequently in the market will be configured as tags for extra convenience, such as consumer groups with a high consumption in baby products. Feature engineering helps you to generate multiple tags at a time. This reduces redundant expressions when you use tags in application modules. In terms of resource utilization, configuring these frequently-used conditions as tags can reduce the pressure on online computing and save costs.



### 26.4.3.3 Components

#### Tag schemes

A tag scheme is used to define the logic of derived tags, including the type, tag configuration, scheduling configuration, and parameter configuration of the tag scheme.

When creating derived tags, you must define the tag generation logic, the algorithms based on existing tags, the result fields corresponding to the new tags, and tag objects to be associated. If the result is a MaxCompute partitioned table, you must also configure which output field is used as the partitioning field.

The task includes the following two scheduling types: one-time schedule and recurring schedule. The tag generation logic supports TQL expressions, common functions, and logical expressions.

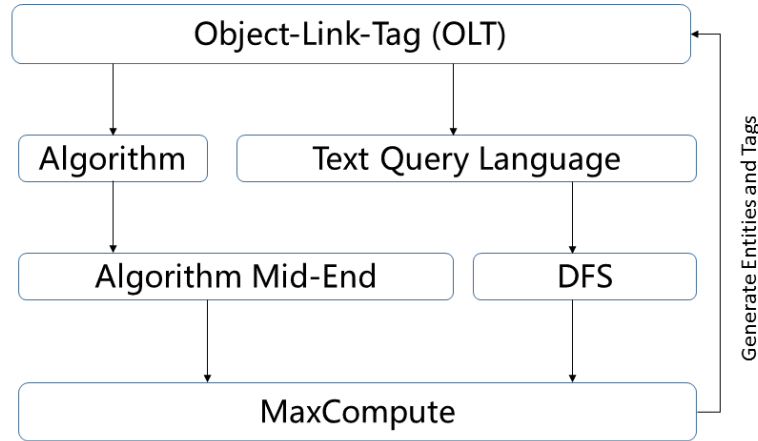
#### Tag tasks

After the tag scheme is configured, you need to run it to generate a new derived tag. On the Tag Tasks page, you can manage the tasks generated by task schemes and schedule these tasks. You can run a tag scheme after it is modified and submitted.

A task will be generated each time when you click Execute to run a tag scheme. You can set execution parameters for a recurring task. After the task starts scheduling, you can view the task instances, running status, and logs for each scheduled task.

### 26.4.3.4 Technical architecture

The following figure shows the technical architecture of Tag Factory.



### 26.4.4 Dashboards

The Dashboards page provides an intuitive data view and analysis reports. You can gain an insight into the analysis results in an intuitive manner.

Based on the created tag models, you can modify datasets, edit tags to be queried, set filtering conditions, and then publish the datasets online by saving these configurations.

You can create a report on the Dashboards page. You can add multiple visuals to a new report. You can also configure a dataset for each visual and set the returned fields of the dataset to the displayed data of the visual. You can configure control parameters for each visual to be displayed in a better way.

On the Report Permissions page, click Role Management tab and click Create Role in the upper-right corner to create a user role and set different role permissions.

On the Report Permissions page, click User Authorization tab and click Edit Role in the Actions column of a user to change the roles for the user.

### 26.4.5 Tag sync

Tag sync is one of the most important functions of processing cross-computing data flow for DataQ - Smart Tag Service. When data is required by the corresponding

data service, the tag center can collect the data distributed across multiple storage systems. The data is then subscribed to the location where the data service needs to compute.

In scenarios that require quick response during synchronization, you need to first subscribe to the API factory.

Tag sync includes sync schedules, sync tasks, and task O&M.

- Sync schedules

When you plan to synchronize data from a source schema to a target schema, you need to configure the sync objects, sync tags, and sync parameters. Afterward, preview and save the sync schedules. After creating a sync schedule, you need to manually start the sync schedule to generate a sync task.

You can generate sync tasks by selecting any of the following two modes: Run Now and Recurring. Each time that you click Start to run a sync schedule, a sync task is generated.

You can set execution parameters for a recurring task. After the task starts scheduling, you can view the task instances, running status, and logs for each scheduled task. If you want to synchronize data from MaxCompute schemas to other schemas, or from other schemas to MaxCompute schemas, you must configure a partitioning column.

- Sync tasks

After a sync schedule is started, a sync task is generated. You can view the task status, scheduling mode, and operations.

- Task O&M

Select a sync schedule, and you can view the detailed logs of the sync schedules instance based on the scheduling mode.

## 26.4.6 Homepage

As the middle layer for business-based data, DataQ - Smart Tag Service accumulates a large number of physical tables associated with tags and collects lots of logs. Tag statistic provides the capabilities of exploring and metering these tables and logs.

Tag statistic also provides the statistics of the following smart data assets:

- The links and total number of tags that are created in the object-link model on the platform.
- The number of analysis APIs configured and the number of times these APIs are called.
- The top tags and objects that are calculated by frequently used tags and their values.
- The derived tags that are mined by analyzing the tag query expressions.

## 26.5 Benefits

DataQ - Smart Tag Service has the following benefits:

- Easy to use

Converts data into systematic business tags, reducing the knowledge requirement to use big data for business personnel.

- Efficient tag production

Builds a tag factory to simplify the tag creation and improve the production efficiency.

- Easy to share

Converts tags into data services that are understandable to the business. This enables data can be easily shared and used.

- Accumulates business tags that are assembled by upper-layer applications and saves these tags to the tag center to form a closed-loop.

## 27 Apsara Bigdata Manager (ABM)

---

### 27.1 What is Apsara Bigdata Manager?

#### Background

**In their daily work, O&M engineers and data service development engineers of Alibaba Cloud big data platform often need to manage various big data products , including offline computing engines, real-time computing engines, analytic and query engines, AI platforms, and big data applications. These big data products are sometimes closely related to each other. Therefore, a one-stop O&M platform for these big data products is urgently required to improve O&M and development efficiency. Against this background, Apsara Bigdata Manager (ABM) is developed.**

#### Supported products

**Currently, ABM supports O&M on the following big data products:**

- **MaxCompute**
- **DataWorks**
- **StreamCompute**
- **Quick BI**
- **Graph Analytics**
- **Elasticsearch**
- **Dataphin**
- **DataHub**
- **Machine Learning Platform for AI**

**ABM supports O&M on the business, services, clusters, and hosts of these big data products. Besides, you can upgrade big data products, customize alert configurations, and view the O&M history in ABM.**

**By using ABM, on-site Apsara Stack engineers can easily manage big data products , such as viewing resource usage, checking alerts and fix methods, and modifying configurations.**

## Challenges

The stability of a big data service may be adversely affected by not only an unstable product platform, but also poor business implementation, such as slow or bad SQL queries, data skews, and log-tail jobs. Therefore, platform O&M engineers alone cannot ensure service stability. Instead, service development engineers and platform O&M engineers must work together to improve service stability.

In addition to platform O&M, ABM is evolving to provide data O&M and even intelligent O&M capabilities. Additionally, it is exploring ways to implement cross-computing engine management and support O&M for more products, including big data applications, computing engines, scheduling systems, storage, operating systems, and networks.

## 27.2 Benefits

Based on a mature O&M mid-end base, Apsara Bigdata Manager (ABM) can quickly connect to big data products and provide comprehensive O&M capabilities for each product.

### O&M mid-end base

In the O&M mid-end base, ABM provides many built-in services and SDKs for constructing business O&M capabilities. Each product can easily connect to ABM and obtain an exclusive O&M site. ABM allows you to construct O&M capabilities in a visualized, configuration-based, and function-based way. This minimizes business customization costs.

In Apsara Stack, the O&M mid-end base of ABM provides the following services:

- **Job platform:** supports visualized job management, execution, and scheduling. This satisfies various needs of visualized O&M.
- **Knowledge graph:** supports storage, integration, and query of data generated in different scenarios. This resolves the difficulties in integrating and querying dispersed data.
- **Function as a service (FaaS):** supports low-cost trial and error, fast business code development, and function-based service logic management. This relieves users from complex project organization, dependency management, deployment, and scaling and allows them to focus on business.

- **Application management:** stores service logic and configurations in a hierarchical way, and supports highly flexible extension capabilities. This allows users to configure complex application structures with simple configurations by using JSON.
- **Inspection service:** provides a universal solution for checker management and scheduling, and supports disparate alert data sources. The inspection service can be embedded into any page of an application site.
- **Third-party system adaptation:** allows users to use one SDK to call APIs of all connected third-party systems.
- **Authorization proxy:** adapts to AAS and OAM in Apsara Stack, provides capabilities such as visualized user management and authorization management, and satisfies the authorization and authentication requirements of third-party systems.
- **Gateway service:** integrates all service APIs so that external systems can call these APIs uniformly. In addition, isolation, decoupling, and scaffold capabilities are provided for authenticating all requests and perform other processing uniformly.
- **Apsara Infrastructure Management Framework synchronization:** adapts to the Apsara Infrastructure Management Framework base of Apsara Stack, and provides encapsulated interfaces for querying and managing full host data.
- **Tunnel service:** uses StarAgent to shield the differences of underlying command execution tunnels and provides universal interfaces. This allows users to deliver commands and files to a large number of hosts, and aggregate and query the statuses of these hosts.

#### Quick business construction

Based on the O&M mid-end base, ABM supports over ten big data products and provides stable and reliable O&M capabilities for these products, including MaxCompute, DataWorks, Realtime Compute, DataHub, and Quick BI.

## 27.3 Architecture



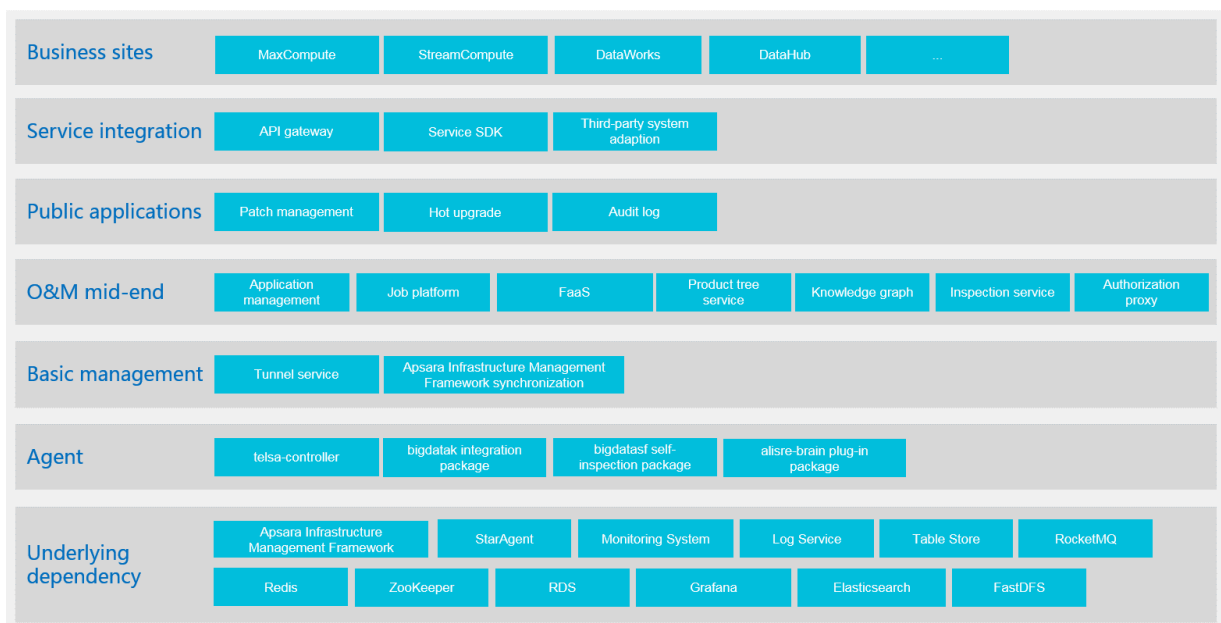
### 27.3.1 System architecture

This topic describes the system architecture of Apsara Bigdata Manager (ABM) and the functions of each component.

ABM uses a microservice architecture that enables data integration, interface integration, and feature integration through a unified platform, and provides standard service interfaces. This architecture enables a consistent user interface , which means the O&M operations are the same for all products. 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.

Figure 27-1: Architecture



#### Underlying dependency

ABM depends on open-source systems from Alibaba and third parties.

- Uses StarAgent and Monitoring System of Alibaba to run remote commands and remote data collection instructions.
- Uses ZooKeeper to coordinate primary and secondary services. This guarantees high availability of services.
- Uses RDS to store metadata, Redis to store cache data, and Table Store to store large amounts of self-test data. This improves service throughput.

## Agent

**The agent provides client SDKs, scripts, and monitoring packages to be deployed on each management host.**

## O&M mid-end and basic management

**The O&M mid-end and basic management components form the base of ABM. Each service in the two components provides its own capabilities for business sites. This enables quick construction of business sites and makes the capabilities of each business site complete.**

## Public applications

**Based on the O&M mid-end, ABM provides multiple public applications. These applications are designed with special purposes and adaptive to all big data products supported by ABM.**

## Service integration

**Service integration functions as a link between business sites and underlying components. It integrates interfaces of all internal services, adapts to various third-party systems, and provides a unified SDK for users.**

## Business sites

**Business sites are constructed based on the O&M mid-end of ABM and cover all big data products, including MaxCompute, Realtime Compute, DataWorks, and DataHub. A business site functions as a one-stop O&M portal of a product.**

# 27.4 Features

## 27.4.1 Small file merging

**This topic describes the small file merging feature of ABM for MaxCompute.**

### What are small files

**Apsara Distributed File System stores data in blocks. The size of each block is 64 MB. Small files in this topic refer to files whose size is less than 64 MB. Reduce computing or real-time data collection through tunnels will generate a large number of small files.**

## Impacts of small files

- **More small files consume more instance resources. In MaxCompute, a single task instance can handle up to 120 small files. Therefore, too many small files cause a resource waste and deteriorate system performance.**
- **Too many small files cause high pressure on Apsara Distributed File System, and decrease the utilization rate of disk space.**
- **Too many small files occupy a large amount of memories of Master servers and Chunkservers in Apsara Distributed File System. When the memory usage exceeds 50% of the safety limit on a Master server of Apsara Distributed File System, the cluster stability is affected.**

## Method of merging small files

ABM uses the MaxCompute SDK to generate merge tasks for merging small files. This method increases merging concurrency to the maximum extent. Currently, you can create merge tasks by cluster or project. You can configure whether to allow merge tasks to run concurrently and specify the start and end time for each merge task.

## 27.4.2 Job snapshot

This topic describes the job snapshot feature of ABM for MaxCompute.

In this topic, all jobs refer to MaxCompute jobs. When a job is executed, ABM saves detailed job logs. These logs are used to generate a job snapshot. The following figure shows an example of the job snapshot page.

Jobid	Project	Quota ...	Submit...	Elapse...	CPU Us...	Memor...	DataW...	Cluster	Status	Start TL...	Priority	Type
201907250837	odps_smoke_tr	odps_quota	ALYUN\$	18Seconds	200(200%/0.64)	2816(275%/0.2)		HYBRIDODPSC	Running	2019-07-25 16	1	CUPID
201907221435	biggraph_inter	biggraph_quot	ALYUN\$	66Hours2Minu	0(0%/0%)	0(0%/0%)		HYBRIDODPSC	Running	2019-07-22 22	1	CUPID

The job snapshot feature supports the following functions:

- **Displays information about current and historical jobs, including the resource usage and queuing status.**

- **Supports aggregating jobs from different dimensions, such as the quota group, submitter, and job status. This allows you to clearly understand the status of current jobs.**
- **Supports generating a detailed Logview page for a single job.**
- **Supports terminating jobs.**

## 28 E-MapReduce (EMR)

---

### 28.1 What is EMR?

EMR is a managed cluster platform that simplifies running big data frameworks, such as Hadoop, Spark, Kafka, and Storm. EMR provides you with one-stop big data processing and analysis services, such as managing clusters, jobs, and data.

### 28.2 Benefits

Compared with user-created clusters, EMR provides you with easy and well-organized methods to manage your clusters. EMR also offers the following benefits:

- **Deep integration**

EMR works seamlessly with other Alibaba Cloud services, such as Object Storage Service (OSS), Message Service (MNS), ApsaraDB for RDS, and MaxCompute. This allows data of these services to be used as the input or output of the Hadoop or Spark services of EMR.

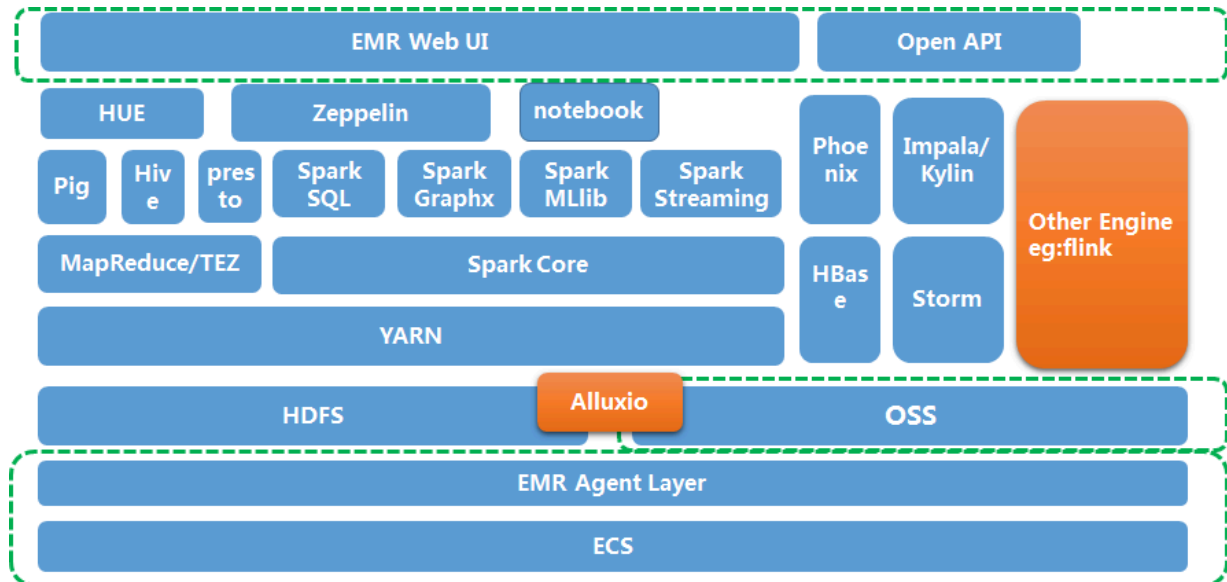
- **Security**

With Resource Access Management (RAM), RAM user accounts are authorized to access different EMR resources.

## 28.3 Architecture

*Figure 28-1: EMR architecture* shows the architecture of EMR.

Figure 28-1: EMR architecture



Dependent on the Hadoop ecosystem, EMR clusters are created based on Alibaba Cloud ECS instances. Clusters work seamlessly with cloud services, such as OSS and ApsaraDB for RDS to exchange data. This facilitates data transit and sharing between services to meet specific business requirements. EMR provides diverse API operations for you to perform actions on clusters, jobs, and execution plans.

For more information about components in the Hadoop ecosystem, see *Terms of Product Introduction*.

## 28.4 Features

### 28.4.1 Clusters

An EMR cluster consists of Hadoop and Spark clusters that each include one or more ECS instances.

EMR allows you to manage clusters in an integrated environment. You can perform centralized management of tasks. These tasks include selecting instance specifications, deploying environment, creating, configuring, running clusters, configuring, running job, and performance monitoring. EMR helps you to avoid complex

scenarios, such as procurement, preparation, and maintenance, and allows you to focus on the deployment and management of your applications.

EMR also offers different combinations of cluster services to meet requirements specific to your business. For tasks, such as daily data statistics and batch compute operations, you can run Hadoop services in EMR. For tasks, such as stream compute and real-time compute, you can run a combination of Hadoop service and Spark services.

## 28.4.2 Jobs

In EMR, you must create a job to run a compute task.

You can create multiple types of jobs in EMR, such as Spark, Hadoop, Hive, Pig, Sqoop, SQL, and shell. You can select a job type specific to your business requirements. Then, you need to specify the job content and configure the Actions on Failures setting.

## 28.4.3 Execution plans

An execution plan includes a set of jobs. You can run an execution plan on an existing EMR cluster or create a temporary cluster to run each job. With scheduling policies, you can run an execution plan at a specified time or on a regular basis. The benefit of an execution plan is that it consumes as many resources as each job requires to maximize resource utilization and reduce costs.

EMR supports the following scheduling policies:

- **Periodic execution:** You must specify the execution interval and start time. Then , an execution plan will run based on the specified schedule.
- **Manual execution:** You must manually run an execution plan.

## 28.4.4 Alerts

E-MapReduce (EMR) supports alerts. You can link execution plans with alert groups. After you configure the Alert Notification setting on the Execution Plan page, contacts included in the specified alert contact group will receive SMS alerts after each execution plan is complete. An SMS message includes the name of an execution plan, cluster name, duration, status of a job, number of successful tasks, and number of failed tasks.

## 29 Quick BI

---

### 29.1 What is Quick BI?

**With the mission to enable everyone to operate as a data analyst, Quick BI provides online ad hoc analysis, drag-and-drop operations, and data visualization. This makes it easy for you to analyze data and monitor business. Quick BI is a tool to view data for business personnel and a booster for digital operations, solving the last mile problem of big data application.**

### 29.2 Benefits

**The overall benefits of Quick BI can be summarized as high compatibility, quick response, powerful capabilities, and user-friendliness.**

#### Rapid data modelling

**You can easily create a dataset with a few clicks. The dataset provides a basic but core function for subsequent data analysis, significantly reducing your reliance on professional capabilities.**

#### Powerful data analysis capabilities

**Quick BI generates professional workbooks. You can perform data conjoint analysis and create reports online. For example, you can create online daily reports, weekly reports, and monthly reports. More than 300 regular data analysis functions allow you to easily acquire business analysis results.**

#### Reliable data access control

**Quick BI adopts an ACL system. An access object is used as a control unit for data permission approval and authorization. Quick BI also incorporates a row-level data permission control solution to realize more fine-grained data access control.**

#### Multiple options for data visualization

**Quick BI provides over 30 components, helping you to achieve effective data visualization. In addition, it supports multi-terminal self-adaptation, allowing you to access multiple terminals in a single operation and significantly improving data analysis efficiency.**



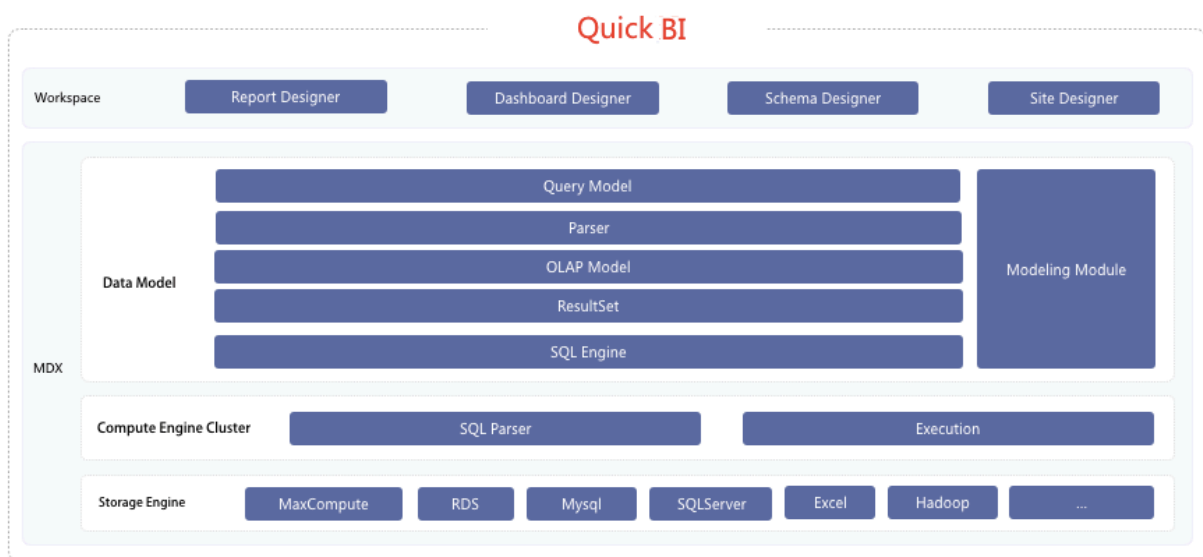
## Multi-user collaboration

**All objects are online. Enterprise users organize their businesses in a shared workspace that allows different team members to operate and analyze data in collaboration. In Quick BI, data analysis can also be realized by joint efforts of multiple members who work on different data analysis processes.**

## 29.3 Product architecture

### 29.3.1 System architecture

The following figure describes the system architecture.

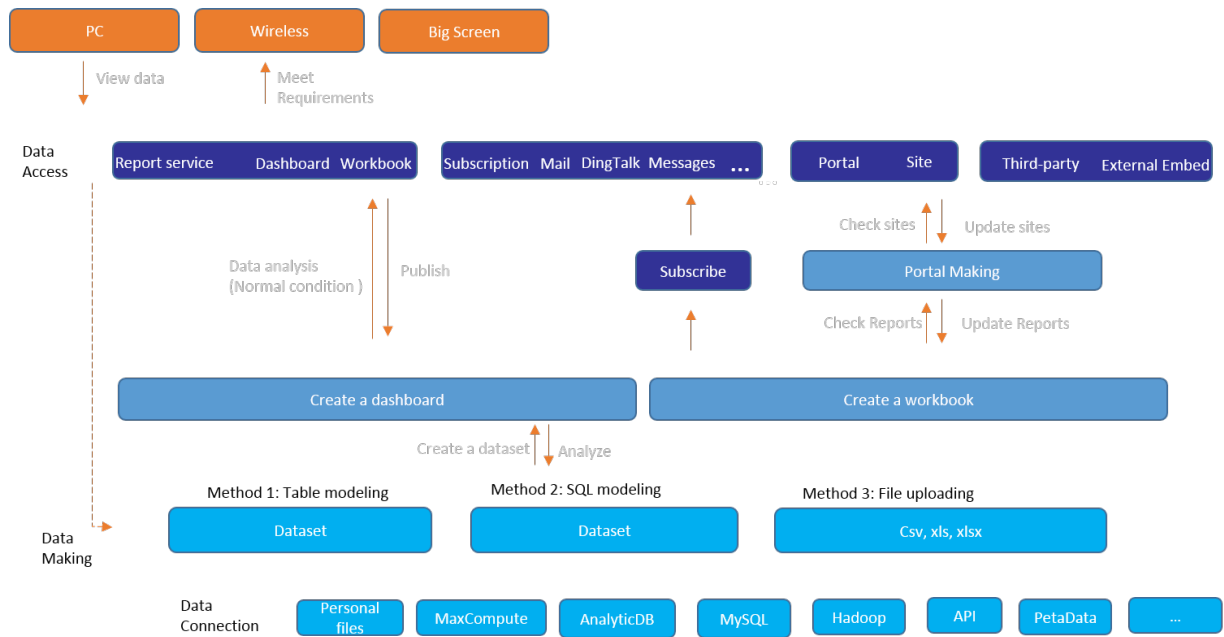


The system architecture of Quick BI consists of four components, including data connection, computing engine, data management, and data analysis. The functions of the four components are listed as follows.

- **Data connection:** establishes a connection with a database to achieve the ability of ad hoc data analysis.
- **Computing engine:** executes SQL queries efficiently in a database based on constructed datasets and user-defined conditions.
- **Data management:** creates datasets that support multidimensional analysis based on tables, SQL statements, and local files, providing dataset management and join operations (star or snowflake schema implementation).
- **Data analysis:** provides dashboard and workbook features to support both enterprise data analysis and ad hoc query.

## 29.3.2 Features

The following figure shows the features and topology of Quick BI.



### Data sources

Quick BI can read data from diverse data sources such as relational databases, MaxCompute, and local files. All data is stored in data sources, and Quick BI does not copy data from the data sources.

### Datasets

Quick BI uses a dataset as the smallest object for data analysis of a project or a specific scenario.

Quick BI datasets support join operations, allowing you to increase the number of dimensions or measures. For example, you can associate a dataset with another dataset to analyze transactions. The codeless process helps you to easily build data objects with powerful functionality.

A date dimension of Quick BI supports multiple granularities, including daily, weekly, monthly, quarterly, yearly, month-to-date, quarter-to-date, and year-to-date.

### Dashboards

Quick BI dashboards provide data visualization functions, and support multi-component filter interaction and a wide range of component settings. Users can use data visualization capabilities as required by their data scenarios.

**Dashboards support mainstream data analysis components, such as vertical bar charts, maps, and funnel charts. It also provides a wide range of function components, such as filter bar, iFrame, and Tab components. Quick BI allows you to design beautiful dashboards with flexible components, reducing your business operation costs and your reliance on specialized staff.**

#### Workbooks

**Workbooks are a unique feature of Quick BI. It offers online data analysis functions in the form of spreadsheet. A cell in a Quick BI workbook is a data unit. Quick BI workbooks allow you to enter and copy data locally, and to retrieve datasets. You can link cells to each other. In addition, workbooks provide the following features:**

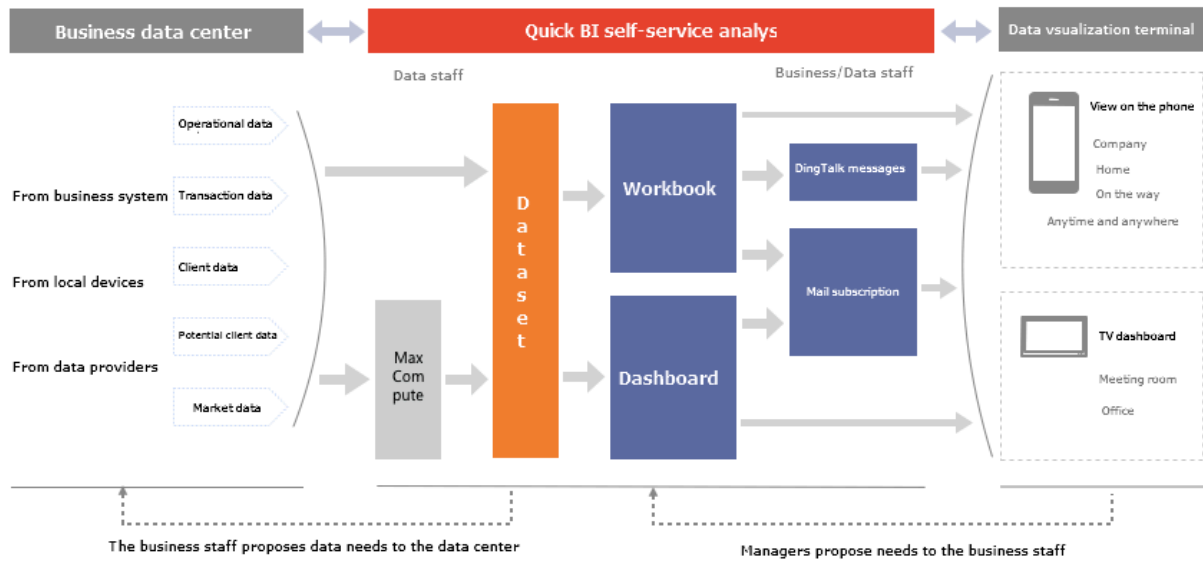
- **Built-in formulas, data aggregation, over 300 functions, and cross-sheet references**
- **Interactive drag-and-drop operations with row and column freezing**
- **Visualization of analyzed data**
- **View, edit, and preview modes**

#### BI portals

**BI portals are directories for dashboards and workbooks based on different scenarios, departments, and marketing plans. This allows you to have a complete view of various business statuses.**

#### Monitoring dashboard

**You can use the single-screen dashboard of Quick BI to create reports and display the dashboard of reports on your TV monitors and other terminals in full screen mode. This allows you to view the status of your business at a single glance.**



### 29.3.3 Deployment

Automatic deployment of Quick BI is carried out through Apsara Infrastructure Management Framework.

Quick BI deployment requires the following resources.

Server role	Specification	Description	Scalability
Quick BI server	Two Docker containers. Each container has 16 GB RAM and 8 cores.	Console page	Scale-out
Quick BI agent	Two Docker containers. Each container has 16 GB RAM and 8 cores.	Computing server	Scale-out
Redis primary node	One docker container. It has 16 GB RAM and 8 cores.	Cache	Scale-out
Redis secondary node	Two docker containers. Each container has 16 GB RAM and 8 cores.	Cache	Scale-out

Server role	Specification	Description	Scalability
Metadata initialization	One docker container. It has 1 GB RAM and 1 core .	Metadata initialization	None
Testing server	One docker container. It has 1 GB RAM and 1 core .	Periodic monitoring system	None

### 29.3.4 Server roles

Quick BI consists of the following server roles:

- **base-biz-yunbi-dbinit:** metadata initialization
- **quickbi-redis-slave:** secondary node of Redis
- **quickbi-redis-master:** primary node of Redis
- **base-biz-yunbi-executor:** Quick BI agent server
- **base-biz-yunbi:** web server in the Quick BI console
- **ServiceTest:** automatic testing server

## 29.4 Features

Quick BI provides the following features:

- Supports a wide range of data sources, such as MaxCompute, relational databases, and local files.
- Supports quick analysis for MaxCompute and Hive data sources. For example, Quick BI can analyze files of total size 100 GB in 10 seconds.
- Provides complete workbooks to enable you to easily make complex reports.
- Enables everyone to easily learn and use.
- Provides you with diverse options for data visualization, and automatically identifies data properties to generate the most appropriate charts.
- Provides strict permission control and adopts multi-factor authentication, which ensures data security.
- Provides an OLAP analysis engine that has comprehensive functions and is still easy to use.

- **Supports collaborative operations, allowing multiple users to analyze data cooperatively.**

## 30 Graph Analytics

---

### 30.1 What is Graph Analytics?

**Graph Analytics is an intelligent visual analysis platform for relationship networks.**

**Graph Analytics is designed to facilitate multi-source data integration, computing applications, visual analytics, and intelligent businesses. Based on relationship networks, Graph Analytics can visualize the properties of objects and reveal the relationship among objects.**

#### Development of Graph Analytics

**Based on years of practice in multiple industries, Graph Analytics has made impressive progresses and evolved with the development of visual analysis and intelligent network analysis. Featuring the OLEP model, Graph Analytics helps users analyze data and relationship networks with ease. Graph Analytics supports data source integration and major compute engines, and can be applied to multiple business scenarios. Graph Analytics aims to build a highly efficient and intelligent platform for network analysis.**

**Graph Analytics provides multiple features, including relationship networks, , search networks, information cubes, intelligent analysis, collaboration and sharing , dynamic modeling, and pattern recognition. With its visual interfaces, Graph Analytics integrates machine computing capabilities with human cognition to provide users with data insight and help users obtain information and knowledge more efficiently.**

**Graph Analytics is oriented to intelligence analysis in the public security, industry and commerce, taxation, customs, banking, insurance, and Internet finance fields. Graph Analytics provides strong support for case analysis, investigations in money-laundering, fraud, and corruption, and correlated transaction cases. This platform helps analysts gain the key information from massive data and find out case-solving clues and valuable intelligence with ease.**

**Graph Analytics has been widely used in Alibaba Group and Ant Financial for risk control, such as anti-fraud, anti-theft, and anti-money laundering solutions.**

## Background and challenges

**Background:** Due to the drastic increase in available information and data, Graph Analytics faces the challenge of obtaining useful intelligence from massive data.

**Challenges:** Receiving massive amounts of complex data from multiple sources, the key challenge for Graph Analytics is to obtain useful information in a highly efficient and intelligent manner.

## 30.2 Benefits

This topic describes the technical advantages of Graph Analytics.

### OLEP model

Graph Analytics developed the OLEP data model based on ontological theories. Unlike the conventional physical data warehouse model that is time and effort consuming, this logical model builds modes for detailed data from multiple data sources based on the object-link-property logic. After you understand and sort the business data, in Administration Console, you can easily configure and define the business logical model, the mapping relationship between the logical model and the physical data, and the application scenario parameters. After you complete these configurations, the business analysis is modeled and defined.

Powered by the OLEP model, Graph Analytics supports real-time deployment and allows the user to add, delete, or modify the graph structure dynamically without cleansing the data.

Graph Analytics integrates data from multiple sources by using the OLEP model. It provides relationship analysis, graph algorithm mining, offline data integration, and other features. Based on the logical mapping between the OLEP model and data, Graph Analytics supports cross-database and cross-engine data integration, so that you can use the same data multiple times with no need to cleanse the data for the relationship analysis.

### Visual interface

Through its visual interactive interfaces, Graph Analytics presents large amounts of data in multiple forms, including relationship networks, map analysis, information cubes, and the behavior chronology pane. This enables users to perform analyses and investigations based on the network, time, and space. Graph Analytics also



**supports various commonly used tools to fit with the operating habits of users and improve the user experience.**

Supports multiple data engines

**Graph Analytics supports multiple data engines. It can handle several exabytes of data and process tens of billions of nodes or links.**

Relationship network models and algorithms

**To optimize data analysis, Graph Analytics provides multiple relationship network models and algorithms for data mining. Graph Analytics combines classic network analysis methods with machine learning and business algorithms to perform intelligent analyses, including overlap analysis, backbone analysis, path analysis, intimacy analysis, and key location analysis. With Graph Analytics, the user can explore complex network data with ease.**

Intelligent network

**Graph Analytics can recognize various intelligent network patterns. It can extract the structure of the graph and use optimized graph algorithms to match subgraphs that have the same structure. With Graph Analytics, users can filter graph structure data from massive data with ease.**

API

**Graph Analytics integrates data from multiple sources by using the OLEP model. It provides relationship analysis, graph algorithm mining, offline data integration, and other features. Based on years of project experience and proven business algorithms, Graph Analytics provides an open API for customized software development. Graph Analytics provides an API for you to call specific operations as needed. The operations cover the following features:**

- **Object and link search**
- **Relationship network powered by years of business experience**
- **GIS service**
- **Label system and the intelligent network**

**You can integrate the network analysis capability of Graph Analytics with your project system, and customize and define the analysis features based on business needs. This API also supports the following features:**

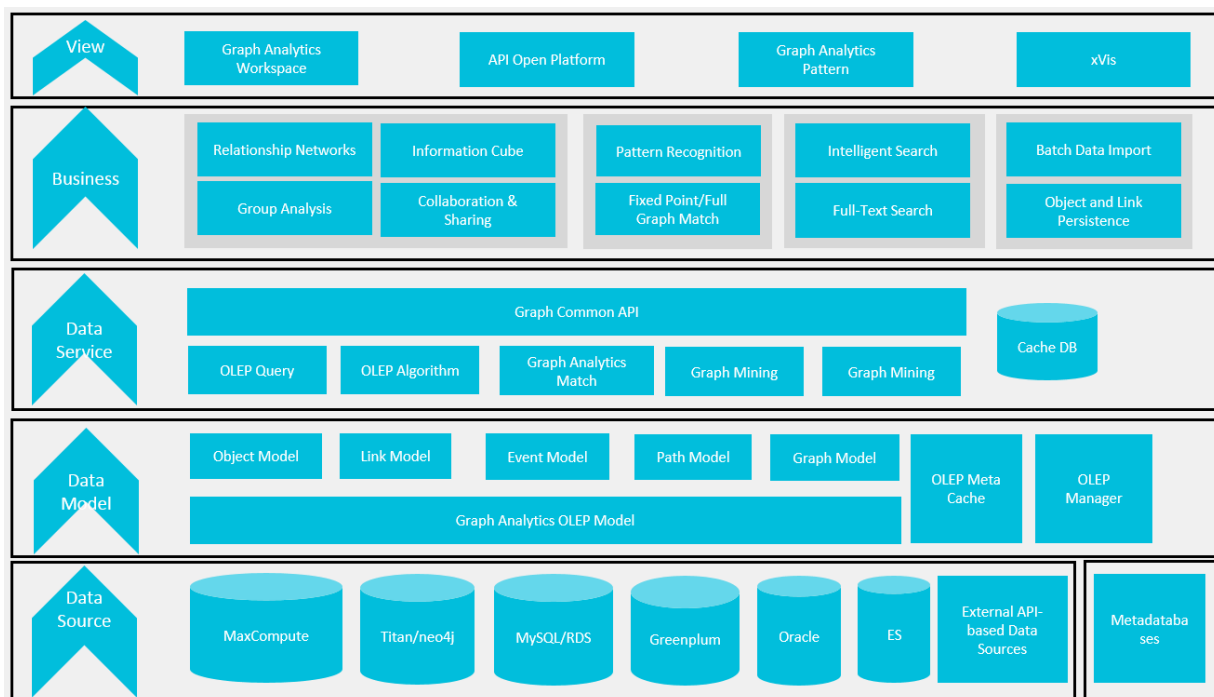
- You can customize API variables to define your own project and run the project efficiently.
- Provided with common page redirects API operations, you can switch between different systems or jump to other pages seamlessly.

## 30.3 Product architecture

### 30.3.1 System architecture

This topic describes the system architecture of Graph Analytics.

Graph Analytics provides multiple components and a multi-layer architecture, including the data source layer, data model layer, data service layer, business layer, and the view layer.



#### Data source layer

Based on the Alibaba Cloud Big Data platform, the data source layer can store and handle petabytes or exabytes of data. It provides powerful data integration, processing, analysis, and computing capabilities. The data source layer provides the following features:

- Supports open source graph databases, such as Titan and Neo4j.
- Supports open source relational databases, such as MySQL, RDS, Oracle, and Greenplum.

- Supports NoSQL databases, including Elasticsearch and KV HBase, a database where each row is a key/value pair.
- Supports external API-based data sources.
- Supports the integration, processing, and online calculation of data from multiple sources.

#### Data model layer

**The data model layer supports the following features:**

- Established based on ontological theories, the OLEP model studies the objects, relationships between natural objects, relationships between social objects, and event information.
- Various types of data are converted into nodes and links in the graph. Based on these nodes and links, Graph Analytics builds paths and graph models to lay the foundation for a subgraph model, providing a standardized data model for data mining and graph algorithm calculation.

#### Data service layer

**The data service layer provides link queries, relationship mining, and graph algorithms for you to analyze relationship networks. This layer supports pattern recognition and extracts graph structure data that is matched with the user-defined graph pattern.**

#### Business layer

**The business layer supports the following features:**

- Graph Analytics provides an API to call application components at the analysis layer. These application components include relationship networks, search networks, information cubes, intelligent judgement, collaboration and sharing, and dynamic modeling.
- Supports intelligent networks, including pattern definition and pattern matching features.

#### View layer

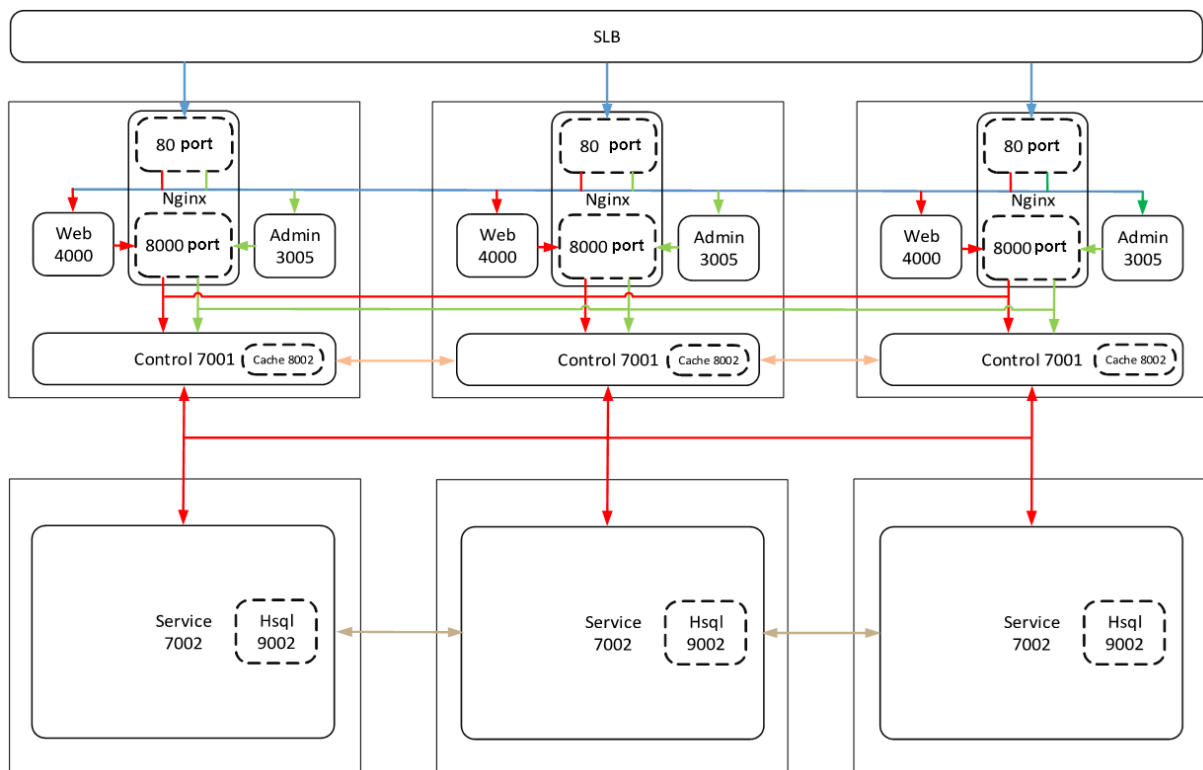
**The view layer refers to the Web layer of Graph Analytics. This layer displays the entire graph, and its features are as follows:**

- Supports multiple layouts of relationship networks to fit with different business scenarios.

- Graph Analytics provides a diversified, visual, and interactive analysis interface and supports various terminals.
- Graph Analytics provides visual components and external APIs and supports third-party system integration.

### 30.3.2 Network architecture

In Graph Analytics, the control nodes and the service nodes are separated. Both single server deployment and clustered server deployment are supported. The network architecture is shown as follows.



#### Reliability and security

Graph Analytics helps users build a reliable and secure system with the following features:

- The Web and the control nodes support stateless load balancing to transfer the load stress from the server where a Node.js failover occurred to another server, without influencing other features.
- Multiple control nodes are deployed. The sessions of these control nodes are synchronized in real time. The servers are stateless, and when one server is down, the other servers can still function, so the user will not be influenced.

- **Uses control nodes to restrict the connection requests to the service nodes, and allocates servers based on the number of users and the usage of resources. When some users query a large amount of data, this operation occupies most of the resources of the current server, but it only influences the current server and will not influence the users on other servers.**

#### Performance and stability

**Graph Analytics improves the performance and stability of a system using the following features:**

- **The underlying layer selects a data source based on the business scenario. Supported by graph database (GDB) and high-performance databases, Graph Analytics allows you to perform a quick analysis on large amounts of data.**
- **The compute layer supports memory databases and allows you to perform the second query and statistics quickly based on the results of the first query stored in the memory database.**
- **Service nodes are allocated based on the available resources and the available number of users to improve system performance and maximize resource utilization.**
- **To ensure a stable system, Graph Analytics uses control nodes to perform traffic throttling and avoid frequent API calls.**

#### Scalability

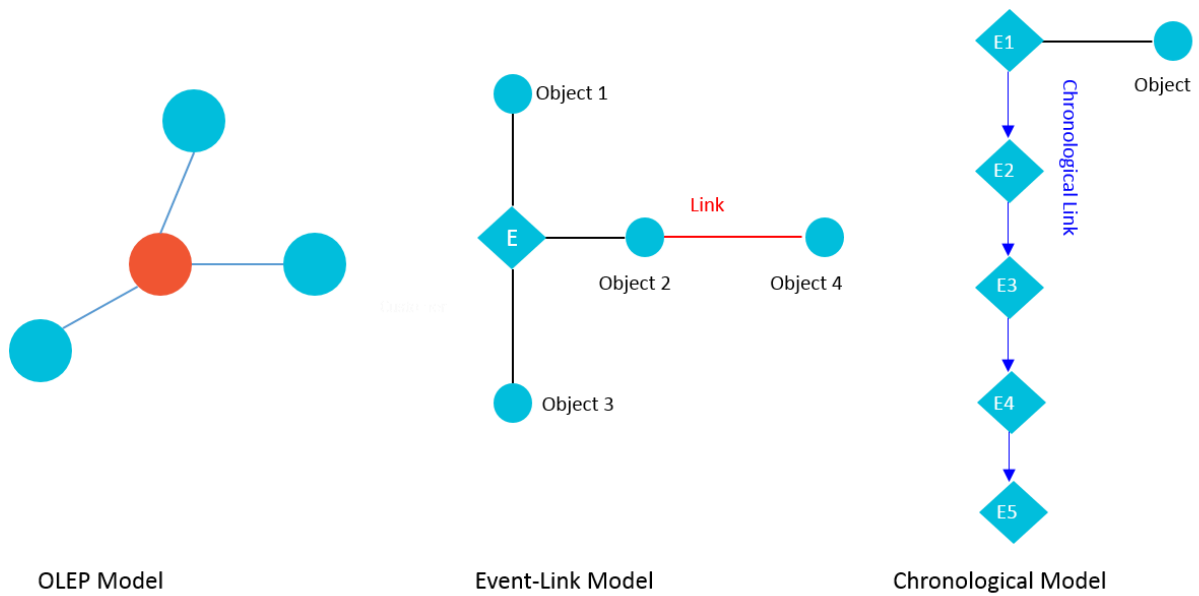
**Graph Analytics supports clustered deployment and horizontal scaling.**

## 30.4 Features and principles

### 30.4.1 OLEP model

**Graph Analytics uses the OLEP model instead of the conventional data warehouse model that requires a large amount of time and effort from the user.**

**The following figure shows how the OLEP model works.**

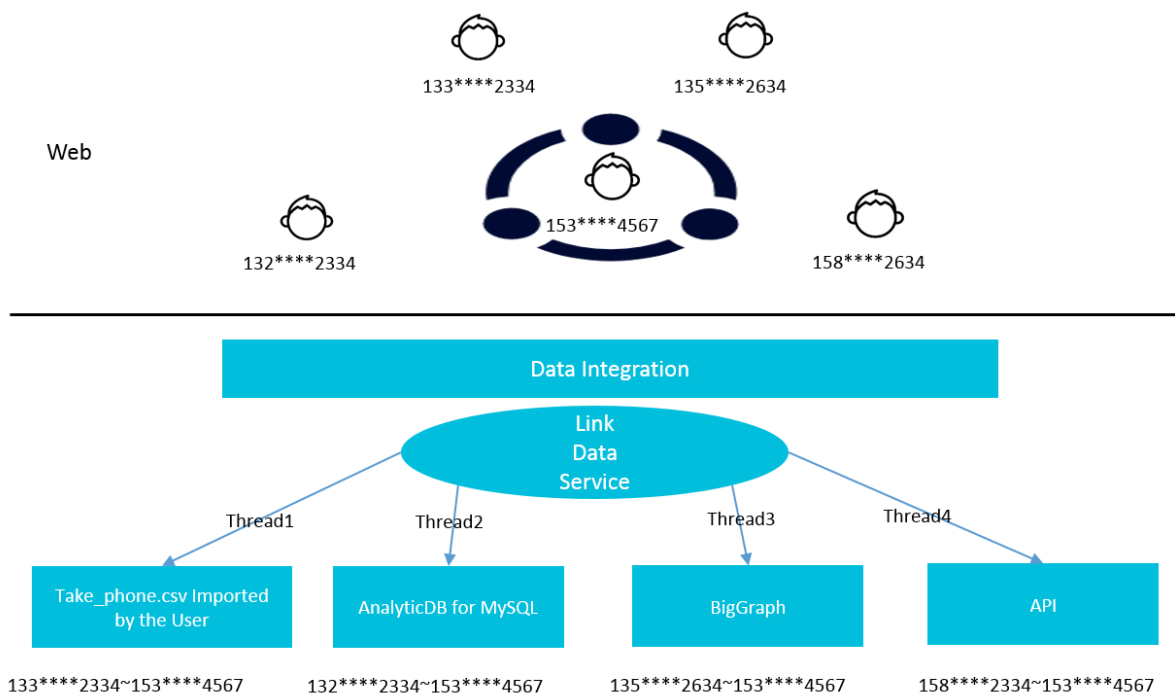


In Graph Analytics, a table can be mapped to multiple objects, links, and events. Columns in the table will be mapped to the properties of objects, links, or events. Based on the OLEP model, every detail record will be mapped to the node, link, and property model of the graph.

### 30.4.2 Data integration

Graph Analytics uses the OLEP model to integrate data retrieved by major search engines.

The following figure shows how Graph Analytics integrates data.



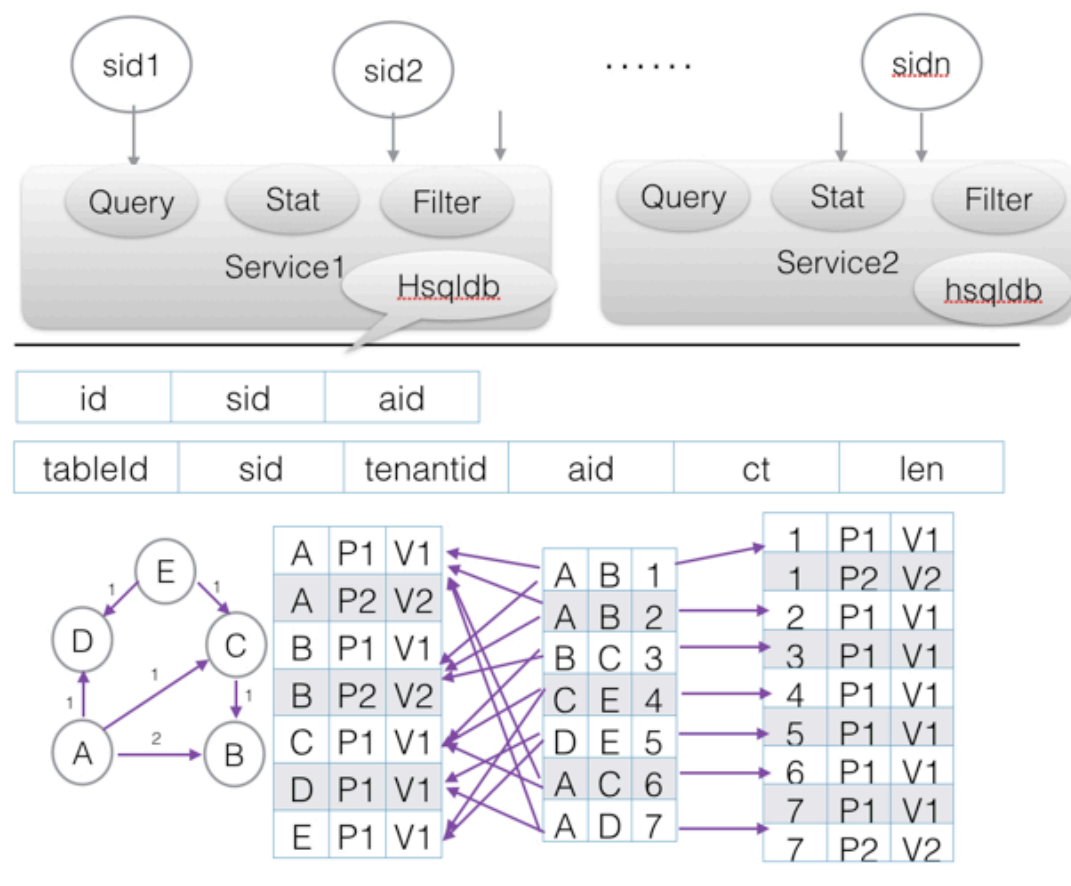
Powered by the OLEP model, Graph Analytics supports major data engines, common SQL syntax, the gremlin language, and API calls.

Graph Analytics supports concurrent processing I/O requests by multiple threads, and simultaneously queries data retrieved by multiple engines that is fit into the OLEP model. Graph Analytics can model and integrate the queried data, and perform visual data analysis based on the graph structure.

### 30.4.3 Separate the graph structure logic from graph details

Graph Analytics separates the graph structure logic and graph details to facilitate large-scale graph analysis.

To separate the graph structure logic and graph details is to store the graph structure in the user's browser while storing the object and link properties in a remote server, as shown in the following figure.



The separation is described as follows:

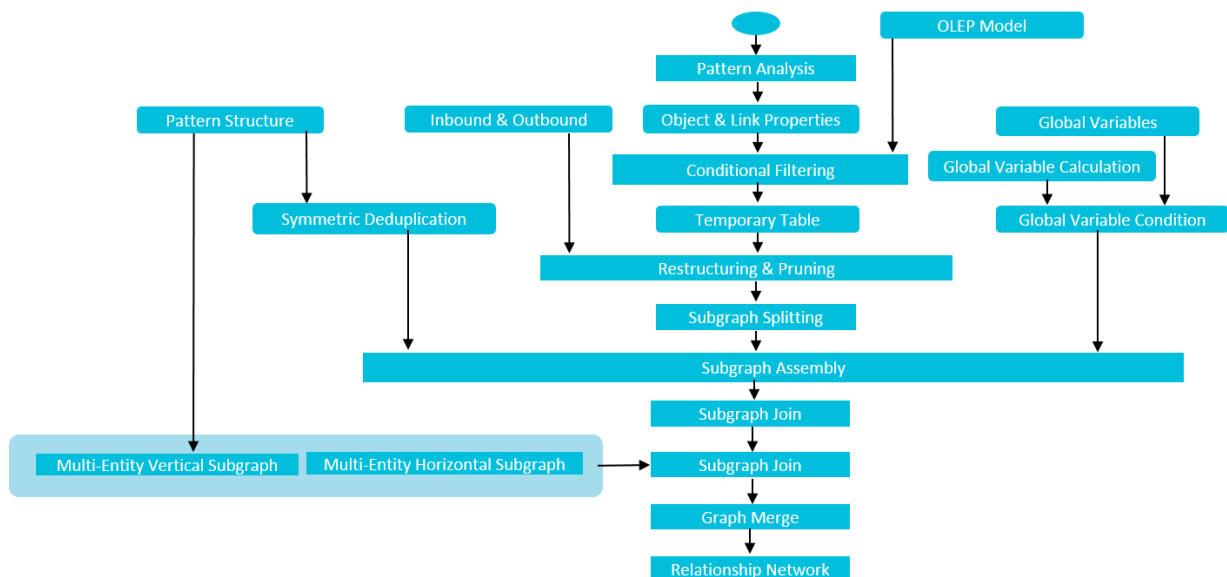
- The graph structure is stored in the user's browser, so the user can arrange the graph layouts and analyze the graph with ease.

- The object and link properties are stored in a remote server. Graph Analytics also has introduced HyperSQL DataBase (HSQLDB) and designed a columnar storage structure to ensure highly-efficient detail queries, statistics, and data filtering. The detailed data is split by analysis. Each user has an independent analysis space with clear structures and can perform queries efficiently.

### 30.4.4 Intelligent network

Based on the intelligent network, Graph Analytics can easily extract the graph structure data that is matched with the user-defined graph pattern from large amounts of data.

The following figure shows how the intelligent network works based on the OLEP model.



The intelligent network performs a pattern matching on correlated data and stores the matched data as intermediate results in a temporary table. Then, Graph Analytics extracts the features of the user-defined pattern, and uses these features as conditions to filter the intermediate results. After filtering, Graph Analytics uses graph algorithms to calculate the amount of relationship data.

Graph Analytics analyzes the overall graph pattern, splits the non-fixed patterns, including linear relationship patterns and intricate relationship patterns, and decomposes the graph into multiple subgraphs starting from the least-related data.

Combining global variables, Graph Analytics analyzes these subgraphs, locates overlapped data in the subgraphs, and merges the data using the SQL Join



**statement multiple times. Subgraphs with non-fixed patterns are merged with these subgraphs in the last step.**

**Queries the key nodes in the pattern, and merges subgraphs that have the same key nodes.**

**Groups the results, queries the relationship network data, and merges the results with the same pattern features.**

## 31 Machine Learning Platform for AI

---

### 31.1 What is machine learning?

**Machine learning is a process of using statistical algorithms to learn large amounts of historical data and generate an empirical model to provide business strategies.**

#### Background

**As a means of production with ever-increasing value, data has been continuously mined by developers and enterprises for valuable information. Machine learning is used to carry out this task and meets user requirements much better than traditional statistical analysis methods. Apsara Stack Machine Learning Platform for AI is developed in line with this technology trend. Machine Learning Platform for AI made its debut in 2015 as the official machine learning platform of Alibaba Cloud. Over the past three years, it has been continuously updated to provide more features, offer optimized user experience, and support more scenarios. It is among the top tier of AI cloud platforms inside and outside China.**

#### Development status

**Machine Learning Platform for AI is a proven, all-in-one platform that provides one-stop platform to implement algorithms and tasks such as data preprocessing, feature engineering, model training, performance evaluation, and offline deployment of production applications. Machine Learning Platform for AI ranks among the top machine learning platforms domestically and internationally, and is used in a variety of sectors such as finance, energy, government, public services, customs, taxation, and the Internet.**

#### Challenges

**Machine Learning Platform for AI aims to lower the barrier to use machine learning algorithms and make AI available in each industry.**

#### Concepts

**Machine Learning Platform for AI is a set of data mining, modeling, and prediction tools. It is developed based on MaxCompute (formerly known as ODPS). Machine Learning Platform for AI supports the following functions:**

- Provides an all-in-one algorithm service covering algorithm development, sharing, model training, deployment, and monitoring.
- Allows you to complete the entire experiment either through the GUI or by running PAI commands. This function is intended for data miners, analysts, algorithm developers, and data explorers.
- In Apsara Stack, Machine Learning Platform for AI runs on MaxCompute. Machine Learning Platform for AI allows you to call algorithms to decouple the applications and compute engines after you have deployed algorithm packages in MaxCompute clusters.
- Provides various algorithms and reliable technical support to resolve service issues. In the Data Technology (DT) era, you can use Machine Learning Platform for AI to implement data-driven services.

Machine Learning Platform for AI can be applied in the following scenarios:

- Marketing: commodity recommendation, user group profiling, and precise advertising.
- Finance: loan delivery prediction, financial risk control, stock trend prediction, and gold price prediction.
- Social network sites (SNS): analysis of microblog fan leaders and social relation chains.
- Text: news classification, keyword extraction, document summarization, and text analysis.
- Unstructured data processing: image classification and image text extraction through optical character recognition (OCR).
- Other prediction cases: rainfall prediction and football match result prediction.

Machine learning can be divided into three types:

- Supervised learning: Each sample has an expected value. You can create a model and map input feature vectors to target values. Typical examples of this learning mode include regression and classification.
- Unsupervised learning: No samples have target values. This learning mode is used to discover potential regular patterns from data. Typical examples of this learning mode include simple clustering.
- Reinforcement learning: This learning mode is complex. A system constantly interacts with the external environment to obtain feedback and determines its

own behaviors to achieve a long-term optimization of targets. Typical examples of this learning mode include AlphaGo and driverless vehicles.

## 31.2 Benefits

Distributed algorithm framework

- **Machine Learning Platform for AI mainly supports three engines: deep learning, parameter server, and MPI.**
- **Deep learning engine with excellent performance.**

Improved model and compilation efficiency

**Collaborative optimization of models and system compilation is a core technology provided by the modern heterogeneous computing infrastructure for AI computing services. Machine Learning Platform for AI supports collaborative optimization of models and system compilation.**

Heterogeneous resource scheduling

**For heterogeneous resources such as GPU resources required by deep learning tasks, an independent cluster is built to schedule heterogeneous computing tasks.**

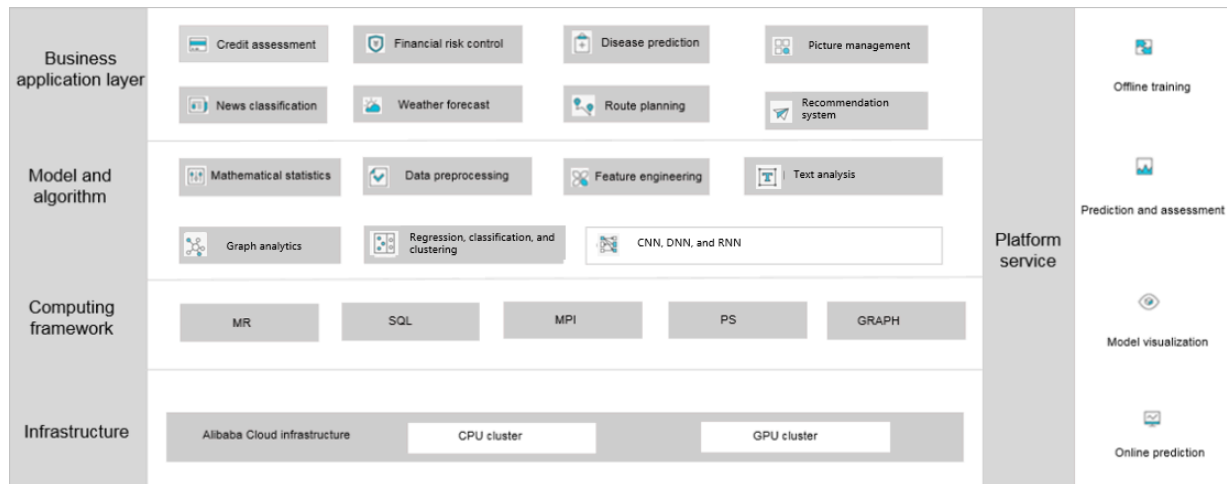
Quality algorithms

**All algorithms come from the Alibaba Group algorithm system and have been tested on petabytes of service data and complex business scenarios. This ensures their sophistication and stability.**

## 31.3 Architecture

### 31.3.1 System architecture

**Machine Learning Platform for AI consists of multiple component systems, as shown in the following figure.**



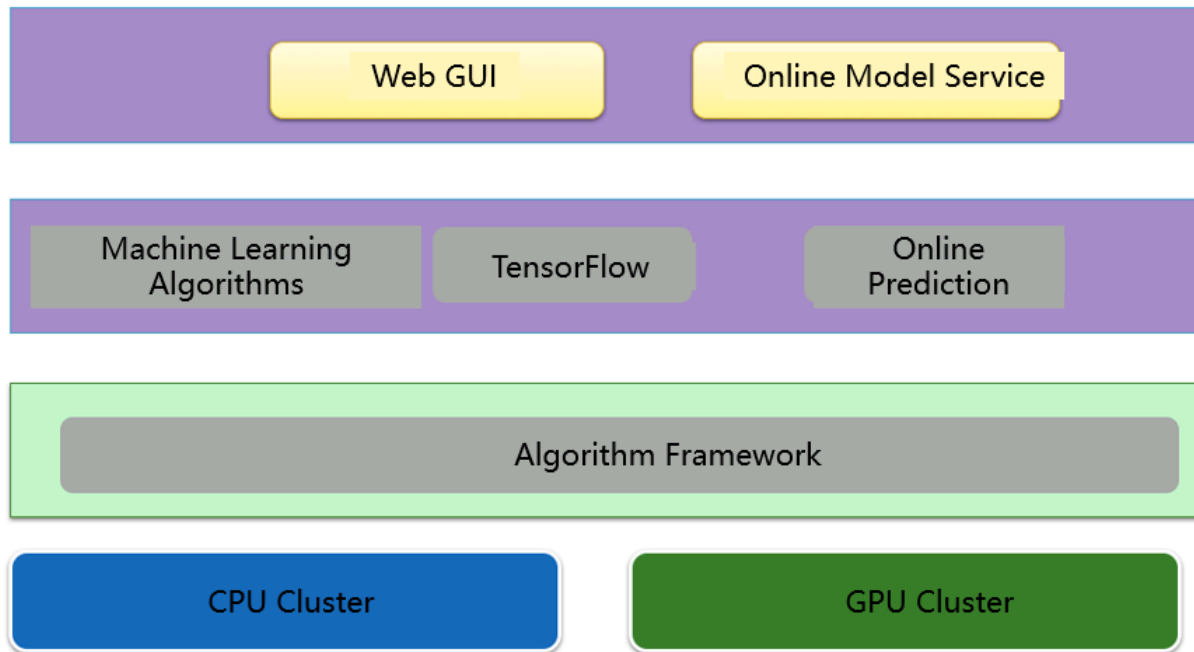
### Architecture of Machine Learning Platform for AI:

- **Infrastructure layer:** includes the CPU and GPU clusters.
- **Computing framework layer:** provides calculation methods such as MapReduce, SQL, and MPI. The distributed computing architecture is used to perform concurrent execution and distribution of computing tasks.
- **Model and algorithm layer:** includes basic components, such as data preprocessing, feature engineering, and machine learning algorithms. All of the algorithm components come from the Alibaba Group algorithm system and have been tested on petabytes of service data.
- **Service application layer:** supports the search system, recommendation system, Ant Financial, and other Alibaba projects in data mining. Machine Learning Platform for AI is applicable in various industries, such as finance, medical care, education, transportation, and security.

If you call models and algorithms in Machine Learning Platform for AI, the system converts the algorithms into compute types. For example, to join two tables, an SQL workflow is automatically generated and then delivered to MaxCompute for calculation and processing. All algorithms are stored in the underlying compute engine as plug-ins for convenient use. This decouples the algorithms from the compute engine.

### 31.3.2 Architecture

Machine Learning Platform for AI consists of the following components:



Component	Description
CPU cluster	The CPU cluster runs machine learning algorithms and provides computing resources such as CPU and memory resources. Computing resources are centrally managed by an algorithm framework. After jobs are submitted, the algorithm framework schedules compute nodes in a CPU cluster and dispatches jobs to the compute nodes.
GPU cluster	<p>The GPU cluster runs deep learning framework jobs and provides computing resources such as GPU and graphics memory resources.</p> <ul style="list-style-type: none"> <li>• Computing resources are centrally managed by an algorithm framework. After jobs are submitted, the algorithm framework schedules compute nodes in the CPU cluster and dispatches jobs to the compute nodes.</li> <li>• For a task that requires multiple workers and GPUs, a virtual network is automatically created to dispatch the jobs to the compute nodes in the virtual network.</li> </ul>
Algorithm framework	<ul style="list-style-type: none"> <li>• The algorithm framework manages CPU and GPU computing resources.</li> <li>• The algorithm framework also provides a basic environment for running algorithms, supporting the Message Passing Interface (MPI) library, MapReduce library, Parameter Server (PS) library, algorithm package, and job isolation by user.</li> </ul>

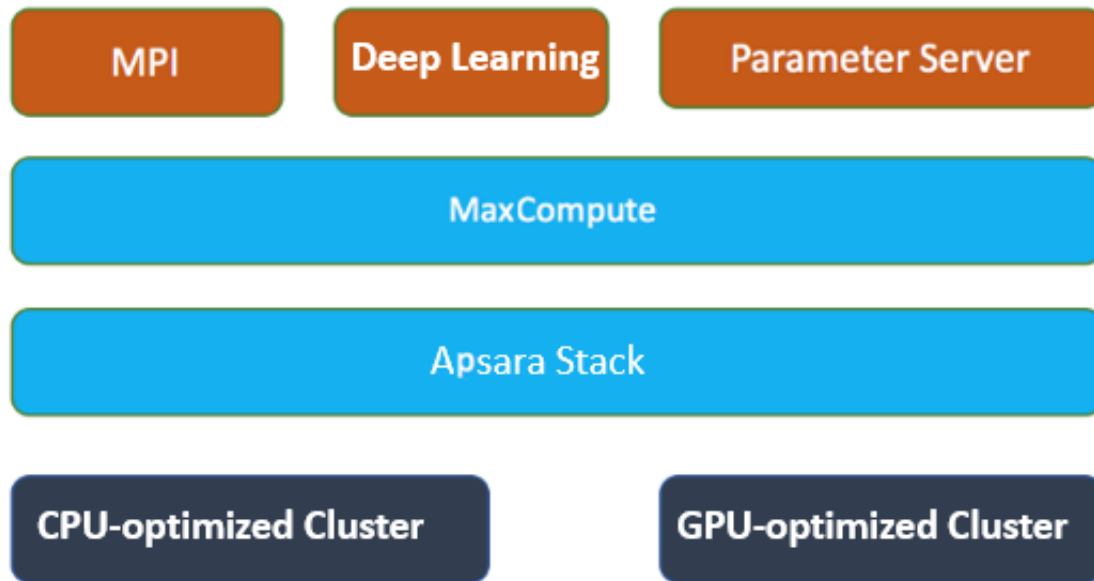
Component	Description
Machine learning algorithms	Machine learning algorithms such as data processing, classification, regression, clustering, text analysis, and network analysis are developed in the basic runtime environment of the algorithm framework. The algorithms are provided as components that can be used in experiments.
TensorFlow	<ul style="list-style-type: none"> <li>Based on the algorithm framework, the deep learning framework TensorFlow is provided. In addition, the performance and throughput of the TensorFlow open-source edition have been improved. The TensorFlow open-source 1.4 edition is supported.</li> <li>You can use TensorFlow to read files from and write models to OSS buckets.</li> <li>When TensorFlow is running, you can start TensorBoard to view the status of parameter convergence during convolution.</li> </ul>
Online model service	You can deploy machine learning models and TensorFlow-generated models as online model services. The online model service supports model version management and blue-green deployment in the rolling upgrade mode.
Web GUI	<p>A visual experiment management console provided by Machine Learning Platform for AI.</p> <p>You can perform the following actions on the Web GUI:</p> <ul style="list-style-type: none"> <li>Create experiments, add algorithm components, and run experiments.</li> <li>You can also deploy models as online model services or publish experiments to the scheduling system in DataWorks.</li> </ul>
Call online model services	Models that are deployed as online model services provide APIs for users to call these services through the Internet.

## 31.4 Functions

### 31.4.1 Resource allocation and task scheduling

Artificial intelligent (AI) tasks typically consume considerable computing resources. Therefore, a distributed system is indispensable. A task must not occupy all resources or occupy a resource exclusively. Instead, a resource is shared by

multiple tenants. Machine Learning Platform for AI balances the efficiency of resource usage between a single task and a cluster.



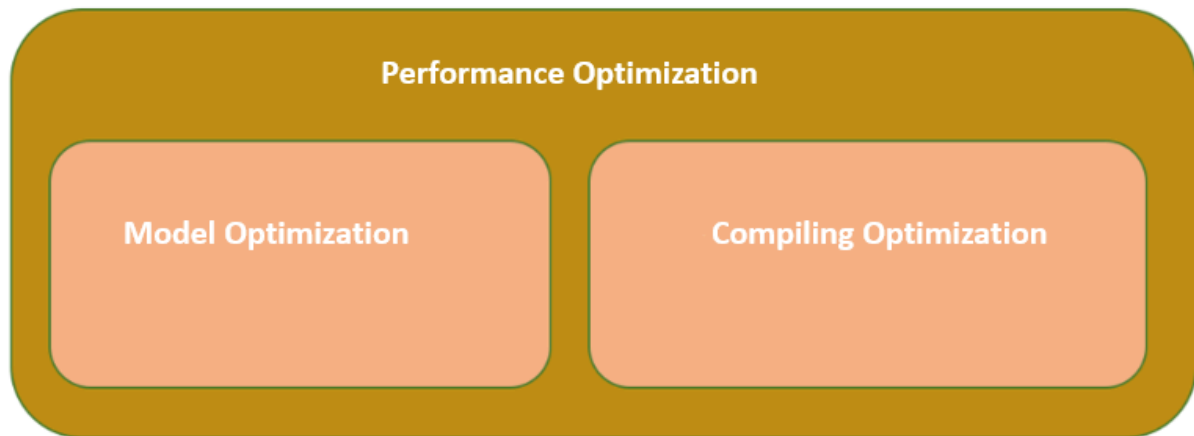
Machine Learning Platform for AI is built on the Apsara operating system and MaxCompute clusters, and is equipped with three types of compute engines: deep learning, parameter server, and MPI. AI tasks and MaxCompute tasks are deployed together to maximize the utilization of resources. For heterogeneous resources such as GPU resources required by deep learning tasks, an independent cluster is built to schedule heterogeneous computing tasks.

To allocate resources to a single task, Machine Learning Platform for AI uses the Tensorflow framework to automatically build a computing chart, allocate CPU and GPU resources, and optimize the task execution efficiency.

### 31.4.2 Model and compilation optimization

Collaborative optimizations of models and system compilation are a core technology provided by the modern heterogeneous computing infrastructure for AI computing services. Machine Learning Platform for AI supports the following types of optimization.





### Model optimization

Many industrial service models are built based on the statistical learning theory. Model parameters can still be regularized and pruned. Besides, the AI-oriented heterogeneous computing tends to implement mixed precision to maximize the computing efficiency while guaranteeing service precision. As the hardware system develops, many technologies have been integrated in Machine Learning Platform for AI. These technologies include low bit quantization, tensor decomposition, network pruning, distillation compression, gradient compression, and hyperparameter optimization.

### Compilation optimization

Model optimization aims to minimize the computing requirements when all service requirements are met. System compilation optimization is used to adapt the specified model to the heterogeneous computing architecture and release the hardware computing resources using end-to-end optimization technologies.

Compilation optimization resolves the following issues:

- **Computing requirement descriptions for service models.** Machine Learning Platform for AI allows you to use advanced abstract languages to describe service models. You need only to describe the computing requirements. The system will translate the descriptions and perform automatic optimization.
- **Hardware system independent computing chart optimization.** Based on the intermediate expression of computing charts, the system implements optimizations that are independent of the hardware system structure. These optimizations include distributed splitting, mixed precision optimization, redundant computing elimination, computing mixing optimization, constant folding, efficient operator rewriting, and storage optimization of computing charts.

- **Optimization and code generation related to the hardware system.** The system performs optimization that is related to the hardware system and generates the target code. The optimization includes storage hierarchy optimization, parallel granularity reconstruction, computing and fetch streaming, assembly instruction optimization, and automatic CodeGen space exploration and tuning.

### 31.4.3 Compute engine

The compute engine provides an advanced programming language for you to compile machine learning models as needed. The engine converts the code into executable tasks at the back end, disassembles or merges the tasks, and submits the tasks to the scheduling system. Machine Learning Platform for AI supports three engines: deep learning, parameter server, and MPI.

#### Deep learning

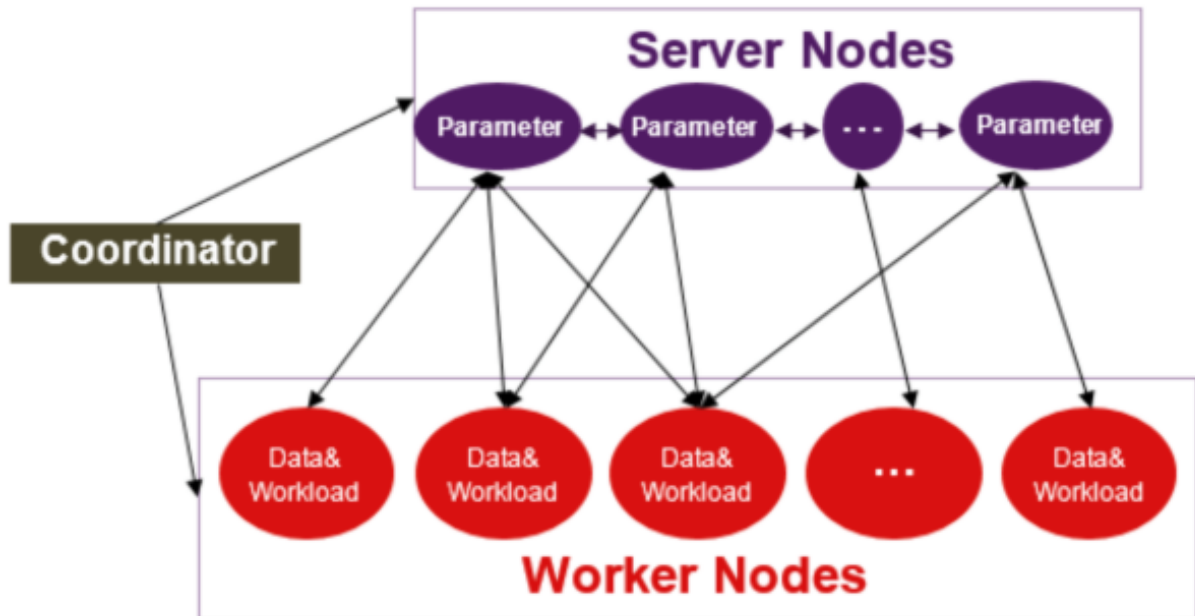
The deep learning engine is developed based on the open-source community TensorFlow. To adapt to the Apsara Stack cluster environment, the following improvements have been made to the deep learning engine:

- **Multiple basic functions are supported.** These functions include image management, service resuming, permission management, and reading and writing MaxCompute and OSS data.
- **The runtime performance of the open-source TensorFlow has been improved.**
  - **Introduces the allreduce network primitive to improve network utilization.**
  - **Replaces the native gRPC mode with the RPC framework for better performance.**
  - **Modifies the synchronization mutex mode to reduce mutex lock competition.**
- **New optimizers and operators are available.**

#### Parameter server

Parameter servers are a type of compute engine provided by Machine Learning Platform for AI for modeling training based on large models and large amounts of sample data. The engine allows algorithm developers to write distributed machine learning algorithm code in the same manner they write standalone code. Algorithm developers can implement distributed machine learning algorithms on the parameter server framework, and verify the algorithms based on tens of

billions of parameter and data dimensions. This shortens the development cycle and allows new algorithms to be released for big data processing.



A parameter server supports the following functions:

- Creates hash indexes for features in real time.
- Allows you to add or delete features.
- Distributed expansion.
- Globally unified checkpoint and exactly once failover.
- Sparse hash feature-based communication.
- Embedding matrix computing based on sparse hash features.

## MPI

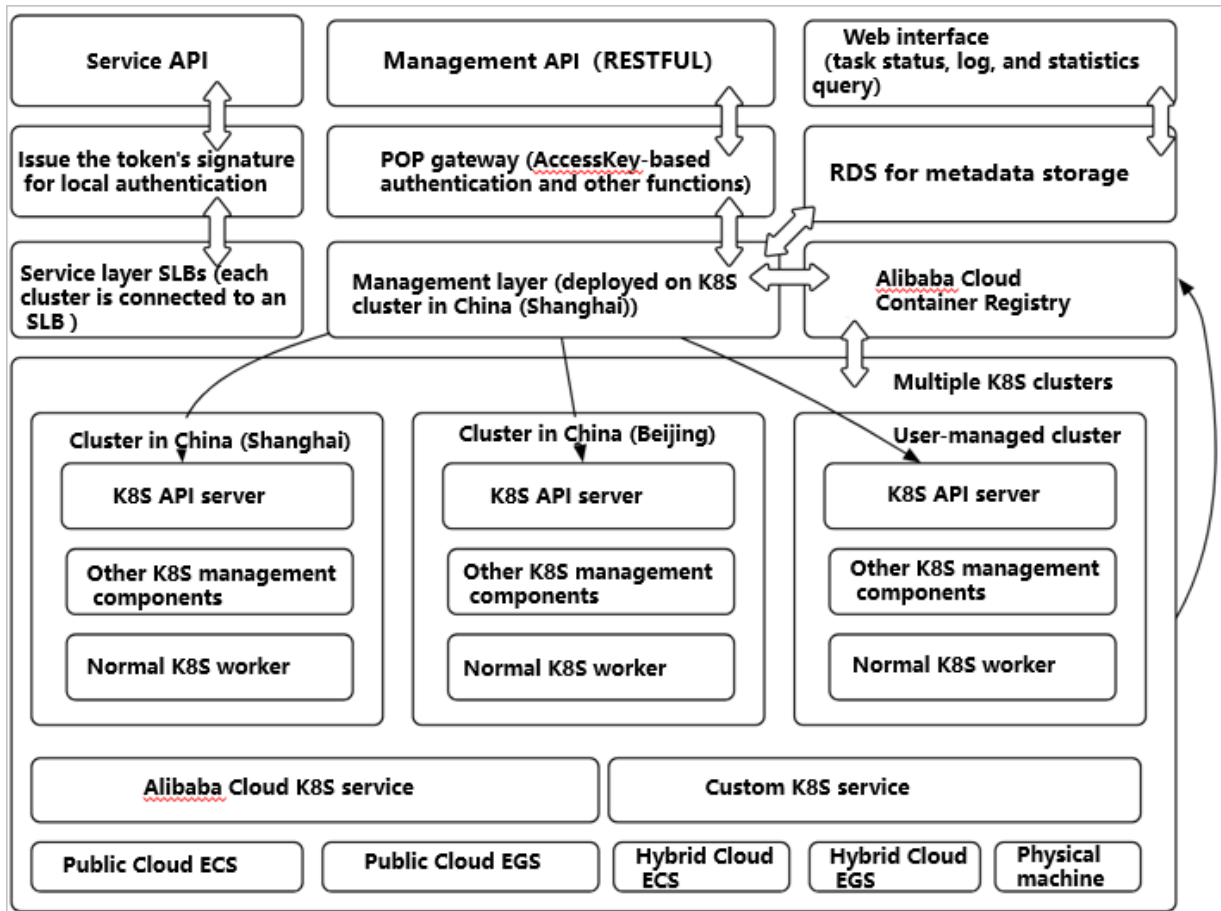
The MPI engine is a generic distributed framework used in the industry. Machine Learning Platform for AI introduces the MPI engine and integrates the MapReduce feature of MaxCompute so that you can implement classic machine learning algorithms such as logistic regression, GBDT, FM, and K-means.

### 31.4.4 Online prediction system

The online prediction system performs predictions tasks in the cloud using multiple types of CPUs and GPUs. The online prediction system is built based on Apsara Stack services, such as ECS, EGS, SLB, and RDS. It uses Docker to manage

resources and isolate resources. It uses open-source Kubernetes (K8s) to schedule tasks.

The overall architecture of the online prediction system is as follows:



The preceding figure shows the architecture of the online prediction system.

API layer

The online prediction service APIs are classified into two types:

- Prediction service APIs
- Prediction request APIs

The two API types are designed with different features to meet different requirements.

- Prediction service APIs are used to create, deploy, delete, and modify prediction services.
- Prediction request APIs are used to process prediction requests sent by clients and return prediction results.

## Computing layer

- **Computing resources**

All computing resources are managed by Kubernetes (K8s). Each node in a K8s cluster is an ECS instance, EGS instance, or physical server.

- **Failover**

Failover depends on the failover solution provided by K8s. K8s allows you to configure a listening service for each container. For example, when the listening port is set to port 80:

- **If an IRP error occurs:**
  1. Port 80 has failed the health check. K8s sets the status of the pod (container group) to Unavailable. Traffic is forwarded to another pod.
  2. When a pod is restarted, the framework initializes the pod and loads the model. Port 80 is not enabled until the model has been loaded.
  3. Port 80 is enabled after the initialization is completed. After port 80 passes the health check at the scheduling layer, the pod is added to the traffic pool again.
- **If a node in a K8s cluster fails:** The keepalive message exchange between the K8s primary node and the failed node fails. K8s sets the status of the failed node to Not Ready. The pod (container group) running on the node is migrated to another node. The traffic is also forwarded to another node.

- **Rolling update**

Rolling updates indicate application updates with zero downtime. The updates are classified into two types:

- **User data update:** User data about the model and processor is updated using an API provided by the online prediction service. The back end creates a new image version based on the current image version and updates the deployment.
- **IRP framework code update:** The framework code is updated by creating a procedure to update all user tasks in the back end. Framework code update

follows the rolling update procedure. Users are not aware of the update procedure.

User data and framework code are decoupled during cluster scheduling and they can be updated separately. The system packages user data and framework code into images of later versions separately and then modifies the description file of the existing application deployment. K8s performs rolling updates for running pods and dynamically switches the traffic to ensure that users are unaware of ongoing updates.

### 31.4.5 List of functions by module

Machine Learning Platform for AI provides a complete workflow of machine learning, such as data uploading, data processing, data visualization, model training, model deployment, model evaluation, and model utilization.

The following table describes the modules and corresponding functions.

Module	Function	Description
Data control	Data uploading	You can upload data through Machine Learning Platform for AI. When you upload data, the data is parsed, verified , and any errors are recorded and reported.
	Data table displaying	On Apsara Stack Machine Learning Platform for AI, click Data Source in the left-side navigation pane to view the uploaded data tables. You can enter a data table name in the search box and click the search icon to search for a data table. Fuzzy search is also supported.
	Data visualization	Right-click a component and choose View Data from the shortcut menu to view data in histograms, pie charts, or line charts.
Model control	Model training	On Machine Learning Platform for AI, click Run in the upper section of the canvas to train and generate a model.

Module	Function	Description
	Model visualization	On Machine Learning Platform for AI, click Models in the left-side navigation pane. Right-click a model and choose Show Model from the shortcut menu to view model parameters. Tree models and linear models can be displayed in tables.
	Model downloading	Right-click a model and choose Export PMML from the shortcut menu to generate and download a PMML file. A PMML is a standard model description file which can be parsed by a variety of open-source software.
	Model-based prediction	You can connect model generation components and prediction components . The system will automatically use the generated model for prediction.
	Model addition, deletion, modification, and query	Right-click a model and choose to add, delete, modify, or query a model.
	Online model service	You can use the online model service to deploy a model and call the corresponding RESTful API for online prediction.
	DataWorks task scheduling	You can deploy experiments to DataStudio as DataWorks tasks and configure the system to periodically run the tasks.
	Model evaluation	You can evaluate models using confusion matrix, binary classification evaluation, clustering model evaluation , and regression model evaluation. Models are evaluated based on metrics such as F1 score, AUC, and KS. All evaluation results can be viewed in tables or charts.
Experiment control	Whole experiment lifecycle control	You can add, delete, modify, query, and copy experiments.
	Experiment visualization	Animated visualizations are used to display the entire procedure by which an experiment runs.

Module	Function	Description
	Notifications	The status of a running experiment is displayed in a prompt in the upper-right corner of the canvas, such as success and error messages.
Deep learning	Multiple deep learning frameworks	Three mainstream deep learning frameworks are supported: TensorFlow, Caffe, and MXNet. With many underlying optimizations, TensorFlow delivers better performance than other open-source frameworks.
	TensorBoard	You can view the training status of each layer in a TensorBoard job in real time and display the results visually.
	Automatic authorization	When the data source of a TensorFlow project is set to OSS, you must obtain permissions on OSS before you can run an experiment. Machine Learning Platform for AI supports automatic authorization, allowing you to obtain the read and write permissions on OSS with a single click.
	Visualized TensorFlow execution settings	The TensorFlow component is added to provide related data source settings, allowing you to run the component visually. On the Tuning tab, you can specify the number of GPUs to run with and implement parallel training with multiple GPUs easily.
	Scheduling	Deep learning jobs can be deployed and periodically executed in DataWorks.
Dashboard	Experiment history chart	You can view the experiment history on the dashboard page.
	Running experiments	You can view running experiments or delete a running experiment to save resources.
	Scheduled tasks	You can view scheduled tasks that have been deployed and add, delete, modify, and query tasks through DataStudio.



Module	Function	Description
Templates on the homepage	Machine Learning Platform for AI provides many built-in experiment templates	The experiment templates can be used for a wide range of scenarios such as product recommendations , news classification, financial risk control, haze prediction, heart disease prediction, agricultural loan delivery , and census. All these cases contain complete data sets and instructions about their use. You can also create your own experiments by using these templates.
Online prediction	Model version management	You can upload multiple versions of a model, configure them to share the same resources, and switch between those versions.
	Blue-green model deployment	The blue-green model deployment function allows you to dynamically change the proportions of the traffic forwarded between different versions of a model.
	Online model debugging	The online debugging function of Machine Learning Platform for AI allows you to debug deployed models online and view the debugging results in real time.

## 31.5 System metrics

Metric	Requirement
Core metrics	<p>Provides typical machine learning algorithms, such as the data preprocessing, feature engineering, statistical analysis, classification, regression, and clustering:</p> <ul style="list-style-type: none"> <li>• Provides model evaluation algorithms.</li> <li>• Provides time series, text analysis, and network analysis algorithms.</li> <li>• Provides deep learning frameworks such as TensorFlow.</li> <li>• Provides the GPU job scheduling capability.</li> <li>• Provides the online model service and allows you to directly deploy models to the online model service.</li> <li>• Provides a visual console to help you use visual components to create experiments.</li> </ul>
Function metrics	<p>Supports reading structured and unstructured data.</p> <ul style="list-style-type: none"> <li>• Supports data sampling and filtering algorithms, such as the random sampling, weighted sampling, and stratified sampling.</li> <li>• Supports data merging algorithms, such as JOIN, UNION, and MERGE.</li> <li>• Supports data preprocessing algorithms, such as splitting, normalization, standardization, KV to Table, Table to KV, and adding ID columns to tables.</li> <li>• Supports the principal component analysis (PCA) algorithm.</li> <li>• Supports feature importance evaluation for linear and random forest models.</li> </ul> <p>Supports the following statistical analysis algorithms: the covariance, empirical probability density chart, whole table statistics, chi-square goodness of fit test, chi-square test of independence, scatter plot, correlation coefficient matrix, two sample T test, single sample T test, normality test, percentile, Pearson coefficient, and histogram.</p>

Metric	Requirement
	<ul style="list-style-type: none"> <li>• Supports the following binary classification algorithms : the Gradient Boosting Decision Tree (GBDT), Linear Support Vector Machine (SVM), and logistic regression.</li> <li>• Supports the following multiclass classification algorithms : K-nearest neighbors (KNN), multiclass classification for logistic regression, random forest, and naive Bayes.</li> <li>• Supports the GBDT, linear regression, PS-SMART regression, and PS linear regression algorithms.</li> <li>• Supports K-means clustering.</li> <li>• Supports the following evaluation algorithms: the binary classification model, regression model, clustering model, multiclass classification, and confusion matrix.</li> </ul>
	<ul style="list-style-type: none"> <li>• Supports the deep learning framework TensorFlow.</li> <li>• Supports TensorBoard.</li> <li>• Supports scheduling a deep-learning job to a GPU server.</li> </ul>
	<p>Supports time series algorithms such as x13_arima and x13_auto_arima.</p>
	<p>Supports the following text algorithms: the word frequency statistics, TF-IDF, parallel latent dirichlet allocation (PLDA ), Word2Vec, word splitting, converting rows, columns, and values to KV pairs, string similarity, deprecated word filtering, text summarization, document similarity, sentence splitting, keyword extraction, ngram-count, semantic vector distance, and pointwise mutual information (PMI).</p>
	<p>Supports the following network analysis algorithms: the K-Core, single-source shortest path, page rank, label propagation clustering, label propagation classification , modularity, maximum connected subgraph, vertex clustering coefficient, edge clustering coefficient, counting triangle, and tree depth.</p>
	<ul style="list-style-type: none"> <li>• Supports the online model service and allows you to deploy machine learning algorithm models or deep-learning models to the service.</li> <li>• Provides an HTTP-based API.</li> </ul>

Metric	Requirement
	<ul style="list-style-type: none"> <li>• Provides the Web-based visual editor, which allows you to create an experiment by dragging and dropping components.</li> <li>• Supports releasing experiments to DataWorks for task scheduling.</li> <li>• Supports experiment and model management.</li> </ul>
Compatibility/ openness	<ul style="list-style-type: none"> <li>• Supports the open-source deep learning framework TensorFlow 1.4.</li> <li>• Supports exporting the PMML file from machine learning models.</li> <li>• Supports the online service model and allows you to deploy the model as an API.</li> </ul>

## 32 Dataphin

---

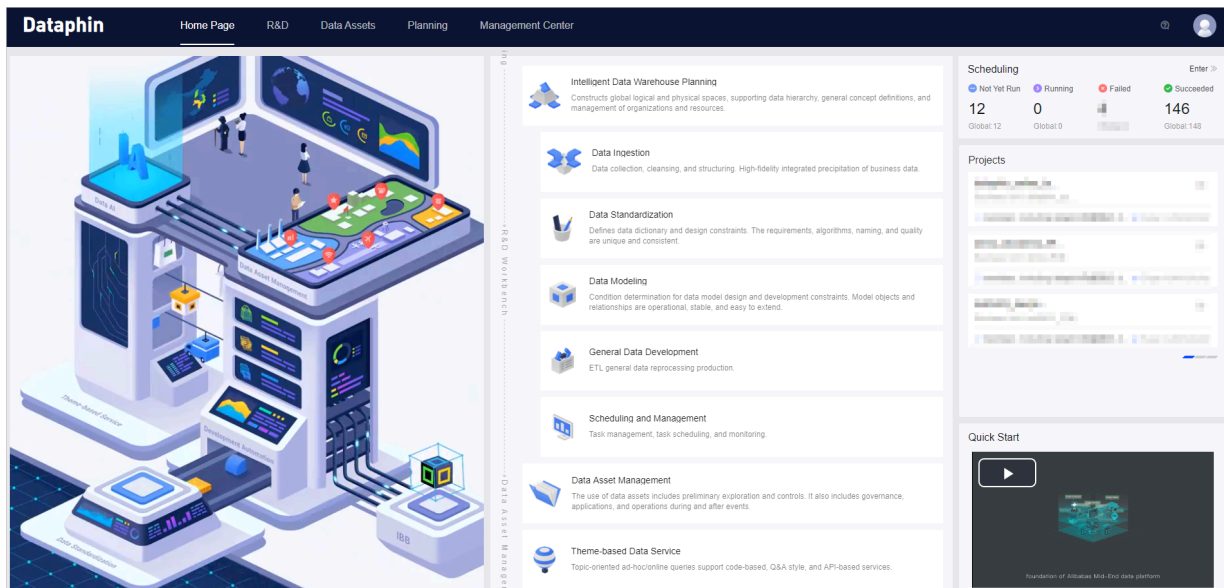
### 32.1 What is Dataphin?

#### 32.1.1 About Dataphin

Dataphin is an engine for creating intelligent big data platforms. It is designed to meet the requirements of big data development, management, and utilization across multiple industries. It adopts an OneData, OneEntity, OneService (product , technology, methodology) big data lifecycle management system. The system is developed by Alibaba Cloud and has been proven by years of practice. Dataphin provides an end-to-end intelligent data creation and management solution covering data ingestion, data standardization, data modeling, data development , data distilling, data asset management, and data services. These features help governments and enterprises build an asset-oriented, service-oriented, closed-loop , and self-optimizing intelligent data system with unified standards to stimulate and drive innovation.

Dataphin is integrated with a large amount of compute and storage environments, which enables you to use a single console to process data from various data sources . By using Dataphin, you can import data, produce standard data by data modeling, and create a tag system by extracting tags from entities. This allows you to generate and manage data assets by using your business data knowledge. Dataphin also provides several types of data services including data table search and intelligent voice search.

The following figure shows the Dataphin R&D workbench.



### 32.1.2 Features

Dataphin has the following modules:

- **Platform**

This module helps you learn more about the entire product system and global settings, and understand the product functions to quickly get started. It also implements system management and control to ensure that all the other modules are running as expected.

- **Global design**

Based on a global view of your business and data, you can design an architecture for your data warehouse. During the design, you need to define namespaces (business units), theme domains (data domains), and terms (global objects). You also need to create projects as management units and add data sources and computing engine sources.

- **Data ingestion**

Based on the projects and physical data sources defined during global design, the data ingestion module supports data extraction. This involves extracting all kinds of data from all business systems and loading the data into the target big data storage. This process achieves data synchronization and integration, which facilitates the building of the source data layer after the integration of data from vertical businesses. In addition, this also provides a solid foundation for further data processing.

- **Data standardization**

Based on the architecture defined in global design and the source data layer built by data ingestion, you can create data elements such as statistical metrics. You can use these data elements to ensure that clear and standardized data will be produced.

- **Modeling**

You can use the data elements created for data standardization to design data models. After the data models are submitted and published, Dataphin automatically generates code for the models and recurring data production tasks. This is a full suite of services that provides complete management of data production on the common dimensional model layer.

- **Coding**

Dataphin provides a code editor for you to configure and submit code tasks.

- **Resource and function management**

Dataphin allows you to manage resource packages (such as JAR type and other file types) to meet data processing requirements. Dataphin supports searching for and using built-in functions. You can also create user-defined functions to meet the specific requirements for functional processing.

- **Scheduling and management**

Dataphin supports policy-based scheduling and management of tasks generated by modeling, coding, and data distilling. The scheduling and management involves data production task deployment, task implementation, dependency checking, and task management. This ensures that all tasks can run as expected and without interruption.

- **Metadata warehouse**

Dataphin allows you to collect, parse, and manage metadata of the source data layer, common dimensional model layer, and distilled data center.

- **Data asset management**

Based on the metadata warehouse, this module supports deep metadata analysis and data asset management. It shows asset distribution and metadata details. This makes it easy for you to search for data assets and learn about data assets in more detail.

- **Ad hoc query**

**This module supports asset data searches through custom SQL queries. You can use the search and analysis engine to quickly search for data in physical tables and theme-based logical tables. Theme-based logical tables are also known as data models or logical models.**

### 32.1.3 Benefits

**Dataphin provides the following benefits.**

- **Data standardization:** The definitions of dimensions, dimension attributes, business processes, and metrics are standardized based on dimensional modeling. This standardization helps to guarantee the quality of data and accuracy of metrics.
- **Efficient and automatic coding:** You can define atomic metrics, business filters, granularity, and statistical periods. By combining these four types of computing logic components, you can then define derived metrics. You can use these components to create data models. Based on your models, the system will automatically generate code to produce data.
- **Optimal intelligent computation:** You can create logical models from business perspectives. After you publish your logical models, the system automatically generates the physical representations of the logical models and the code of the logical models. This reduces your dependence on professional data developers.
- **End-to-end development:** Data ingestion, modeling, development, management, data search, and exploration are combined to implement centralized and efficient development.
- **Systematic data catalog:** Based on standardized modeling, efficient and automatic metadata extraction, Dataphin provides a standardized and user-readable data catalog. The data catalog allows you to spend less time finding the data you require.
- **Efficient data search:** An overview of data assets is provided based on your metadata and data from the Dataphin system database to achieve simple, fast, and intelligent search of data tables and data.
- **Visualized data assets:** A business data asset map (data catalog) is built to help represent your business system from different data perspectives, extract business data knowledge, and learn more about key business stages and data.



- **Easy and reliable data utilization:** Data elements can be used for data production after they are created. You can easily search and access logical tables created based on business themes. This simplifies about 80% of query code.
- **High efficiency:** Dataphin provides end-to-end and intelligent data construction and management tools. This reduces data development requirements. Developers can independently run the extract, transform, and load (ETL) procedure to quickly meet the demand for data. The OneData, OneEntity, and OneService methodology (patent pending) enables the abstraction and definition of models and metrics, automatic coding, automatic theme-based data aggregation and output.
- **Low costs:** Dataphin is metadata-based and algorithm intelligence-driven. Automatic data production is independently performed on both the physical platform (backend computing engine) and logical plane (UI). In addition to comprehensive analysis, tracking, and optimization for data assets, Dataphin ensures optimal computation and storage resource allocation. This greatly reduces the cost of data utilization.

## 32.2 Technical advantages

- **Data standardization and automatic coding:**
  - **Data standardization:** You can define dimensions, business processes, and metrics based on dimensional modeling and by using functions. You can combine these computing logic components to create data models. Then, Dataphin automatically generates code to produce data.
  - **Efficient and automatic coding:** To achieve optimal computing and storage performance, Dataphin optimizes the combination of computing logic components by physically or logically partitioning physical tables referenced by a logical model and generates code for producing data.
  - **Automatic scheduling:** By analyzing code and following scheduling configuration, Dataphin designs a scheduling directed acyclic graph (DAG) to runs tasks in the optimal sequence. This guarantees the performance of data production.
  - **Various models:** By using Dataphin, you can create logical tables in seconds . After you submit logical tables, Dataphin generates and optimizes code in minutes. Dataphin supports common, hierarchy, enumeration, and virtual dimension-based models, transaction and periodic snapshot fact-based

models, and native atomic and composite metrics. Dataphin allows you to use business filters and metrics to create logical tables in the snowflake schema or star schema.

- **Systematic data directories**
  - **Metadata extraction:** Dataphin automatically extracts metadata from logical tables and code.
  - **Metadata parsing:** Dataphin automatically parses metadata to accumulate data assets based on the data asset model.
  - **Visualized data assets:** Dataphin displays the overview, flow, and structure of data assets so that you can learn about data in relevant business scenarios.
- **Easy and reliable use of data**
  - **Logical table-based query:** Dataphin allows you to query required data based on [Logical model. Dimension-associated field. Dimension-associated field. ... Attribute], such as Order. Buyer. Membership type. Type. This can shorten the length of SQL statements by 60%.
  - **Code optimization:** After you submit logical tables, Dataphin optimizes code to achieve optimal computing performance. This improves the coding efficiency and saves resources if compared with physical table-based query.
- **All-in-one development**
  - Dataphin integrates data ingestion, data modeling, data development, scheduling and management, and data search and exploration to develop data in a centralized and efficient way.
  - The Dataphin code editor provides syntax prompts and integrates with metadata for you to track table details with one click.
  - At most 100 members can use Dataphin at the same time. You can grant permissions to members on fields and tables. Different roles have different permissions. Members can apply for permissions as required.
- **Efficient scheduling:** Dataphin can schedule millions of tasks at hourly intervals. After you configure resource settings for tasks, Dataphin can allocate resources to tasks based on your settings.
- **Disparate data sources:** Dataphin can read data from and write data to data sources of the following types: MySQL, Oracle, SQL Server, PostgreSQL,

AnalyticDB, Distributed Relational Database Service (DRDS), MaxCompute, FTP, Hive, and Vertica. It also supports data cleansing and throttling.

- **Multiple computing engines:** Dataphin supports two computing engines: MaxCompute and Hadoop 2.6.0-cdh5.11.2.
- **Deep data distilling (not available yet):** Dataphin focuses on people-related IDs and tags. You can define objects and set computing parameters to identify IDs related to target objects, map IDs of different types, and create tags for target objects. This facilitates the construction of a data management platform (DMP) for marketing.

## 32.3 Product architecture

### 32.3.1 System architecture

Dataphin is deployed in the business system as shown in [Figure 32-1: System architecture](#).

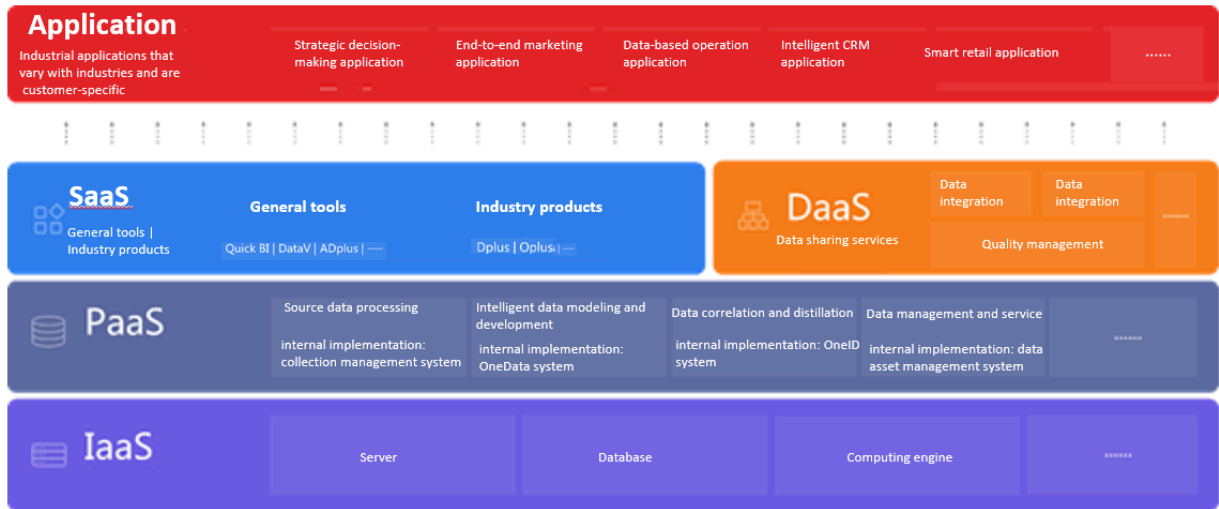
Figure 32-1: System architecture



As the PaaS source data layer of the data platform for business systems, Dataphin can help you to quickly generate easy-to-use data services that can support various data products that consume these services. This implements data-driven business operations. As the source data layer of the data platform, Dataphin can be compatible with different hardware facilities and supports various applications that consumes the platform, forming a data path from IaaS to SaaS, and generating

**business-based standards, ID identification and behavior analysis based on OneID methodology, and manageable and searchable data.**

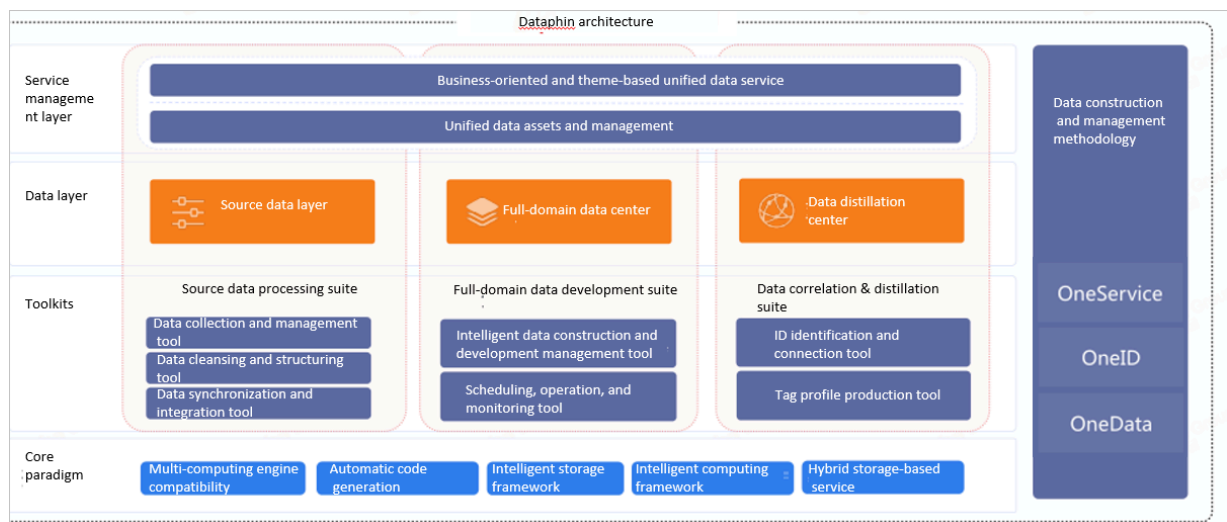
Figure 32-2: Business systems



### 32.3.2 Technology architecture

*Figure 32-3: Product architecture* shows the relationships between the products, components, and architectures in Dataphin.

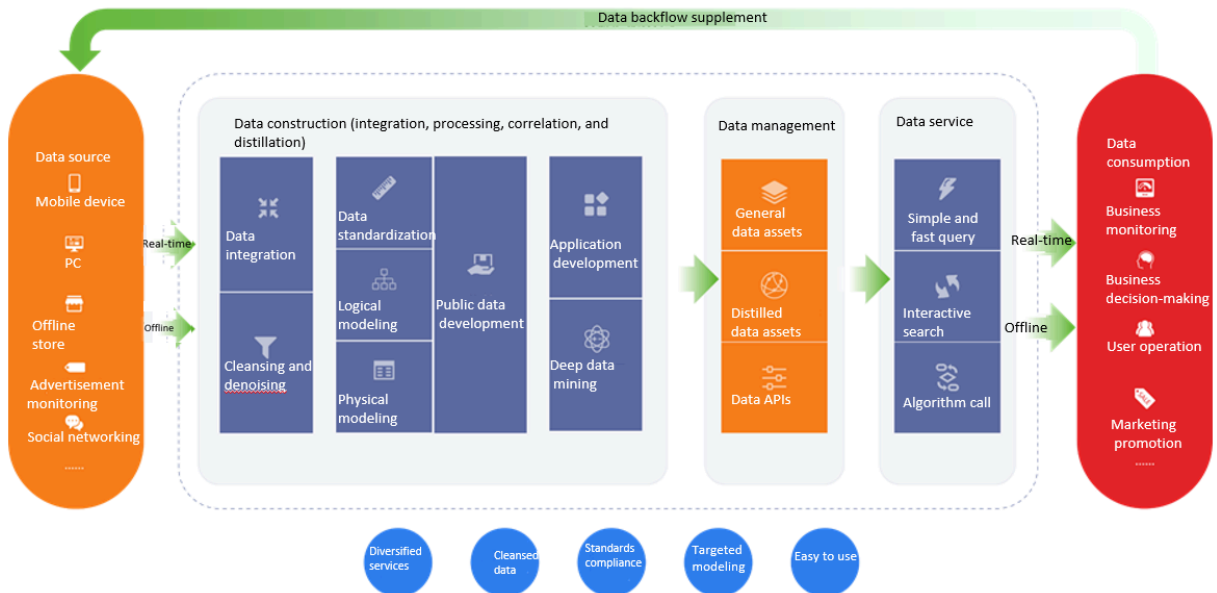
Figure 32-3: Product architecture



**Based on the data construction and management methodology of OneData, OneID, and OneService, the Dataphin system consists of four parts: core paradigm, toolkits**

, data layer, and service management layer. Collaboration between the four parts enables controllable data flows.

Figure 32-4: Data flow



- **Core paradigm:** A technical framework that glosses over the differences of underlying computing, storage, and software systems. This realizes the compatibility of data development with multiple computing engines and improves computing efficiency. It also supports development automation for intelligent storage and computing, and hybrid storage for data service.
- **Toolkits:** This provides data construction and management tools for developers, including standardization and ingestion of source data, data standardization of common data modelling, intelligent modeling development, task scheduling and management, machine learning, connection verification through ID distillation, and tag creation.
- **Data layer:** This layer utilizes core paradigms and tools to output three types of structured data: source data, common dimension model, and application data. These data types help to construct a business-oriented source data layer that provides precise business data, a theme-based common data modeling layer that provides modeling data, and an entity-centered extraction data center that provides deeply-processed behavioral data and tags.
- **Service management layer:** This layer provides insights into data and data service management, allowing both the developers and sales team to obtain high-quality and unified data assets. From a business perspective, the existing

**data is packaged and processed into theme-based data services to guarantee that business data can be centrally searched for and utilized.**

## 32.4 Features

### 32.4.1 Console

**As the basis of Dataphin, the Dataphin console guarantees that all Dataphin members can develop data in a controllable, orderly, and smooth manner. In this console, you can configure global settings, such as account management and computing management. The Dataphin console supports both Chinese and English, and provides introductions and entrances to various modules on its homepage. This helps the super administrator get the whole picture of Dataphin and members of other roles quickly access modules.**

#### Account management

**The Dataphin console allows you to manage member accounts to guarantee secure use of Dataphin. You can connect your enterprise account system to Dataphin. Then , the users who need to use Dataphin can be added to Dataphin as members. Users with the highest privileges can manage accounts and permissions of other users.**

#### Computing management

- **As a Platform as a Service (PaaS) product, Dataphin enables you to select a computing engine type and configure connection settings for your data sources . This makes Dataphin compatible with various environments at the Infrastructure as a Service (IaaS) layer. In this way, Dataphin can develop and compute data in a uniform and stable manner.**
- **Dataphin supports two major types of computing engines: MaxCompute and Hadoop. Dataphin can automatically collect and parse the metadata of these two types of engines. For more information about how to collect and deploy metadata, see [Metadata center](#).**

#### Homepage

- **The Dataphin console provides shortcuts to functional modules, projects, and the scheduling center on the homepage. You can also find an overview of the scheduling center and projects on the homepage.**

- The homepage classifies modules based on the workflow in Dataphin, which consists of data warehouse planning, data R&D, data asset management, and theme-based data services. The workflow helps you learn about product features before you get started, and enables you to quickly access specific modules.

#### Language

To help users from different countries and regions use Dataphin, the Dataphin console selects Chinese or English based on the language of your operating system.

### 32.4.2 Global design

The design of data architecture from the business perspective is a fundamental step in data construction. The architectural design ensures that data is manageable and controllable. The data systems defined and designed during data development, distilling, and management meet mid- and long-term business requirements. The retrieved business data is service-oriented, theme-based, and easy to use.

This module includes:

- Business architecture definition based on business characteristics: business unit maintenance and permission control, data domain definition maintenance and permission control, and defined global statistical period settings and management
- Project definition based on independent data management and collaborative development needs: management of basic project information and computing resource configurations, and member management
- Data source definition based on project computing resources and business data requirements: Data source configuration management

#### Business architecture

The business architecture defines logical namespaces, theme categories, and terminology based on business characteristics to standardize data definitions during architectural design management and data construction control.

#### Projects

A project is a physical namespace defined for resource isolation, user member grouping, and data construction constraint configuration based on the requirements of data development and management teams for independent management of data development projects and efficient management of data resource quality.

## Physical data sources

**Dataphin supports data source creation, modification, and other operations to register and cancel the registration of the required databases. The data source types supported by Dataphin include MaxCompute, MySQL, SQL Server, and PostgreSQL. Data sources are the sources or targets used for data synchronization. Some special types of data sources (such as MaxCompute) can be used as computing engines for projects and function as the computation and storage base.**

### 32.4.3 Data ingestion

**This module selects the required business data for storage based on the source data layer design in the global architecture of enterprise data. In addition, data ingestion module also formulates data synchronization, cleansing, and structuring policies based on requirements for data storage, timeliness, and quality.**

**As an initial stage in data construction, the data synchronization suite is developed based on Alibaba's years of practice in the synchronization and exchange of business data, log data, and other types of data. This achieves high efficiency of raw business data ingestion. Through the pipeline, it can support metadata transmission, acquisition and statistics, as well as simple rule checking and custom fault-tolerant mechanism for data transmission volume and content. This can achieve flexible management and high-quality of data synchronization.**

## Data source configuration

**This module supports data source access and management. The data source list allows you to manage accessed data sources conveniently and add data sources of various types. Currently, the data synchronization center supports data sources including MaxCompute, MySQL, SQL Server, PostgreSQL, and Hive.**

## Data sync

**This module allows you to select source data and target data, configure parameters for incremental or full data synchronization, identify the mappings between source data fields and target data fields, configure transmission traffic and concurrent transmissions, and schedule task nodes after creation.**



## 32.4.4 Data standardization

### Overview

In most cases that involve traditional development, specific and important data creation and development (such as data modeling and metric definition), depend on the developer's professional capabilities. Without a uniform naming convention, standards for development and designs are transferred based on individual and changing documents. This may cause a series of problems such as metric name conflicts or repeated calculation.

Based on the OneData methodology, Dataphin standardizes the definition of important data elements such as dimensions, business processes, and metrics. This ensures unique computing logic and names, and eliminates metric ambiguities during the initial stages of architectural design. In addition, Dataphin provides form-based interfaces for you to create multiple metrics at a time. This lowers the requirements of data development and increases overall development efficiency. This also allows business users with limited data analysis expertise to carry out development work by using Dataphin.

Data standardization involves defining five types of data elements: dimensions, business processes, atomic metrics, business filters, and derived metrics.

### Dimensions

- A dimension is unique within a business unit and it exclusively belongs to a data domain. This standardizes naming and theme classification.
- You can create dimensions by adding additional attributes to an existing dimension, which is used as a parent dimension.
- Dataphin supports the creation of various types of dimensions, including common, common (hierarchy), enumeration, and virtual dimensions.
- Dataphin allows you to view and manage the list of dimensions created in a specific business unit or a specific project. You can also view and modify each dimension.

### Business processes

A business process is a collection of the smallest unit of behaviors or events that occur in a business activity. For example, the smallest unit of behavior can be to create an order or browse a web page. The behaviors occurring in a business

process, such as paying for an order and browsing a web page, are recorded in a fact table. The fact table models a particular business process.

Similar to dimension, business process is a key concept in the OneData methodology used for designing the data architecture. It works with dimensions to define the data architecture. Dataphin supports standard definition for business processes. This allows you to check the overall business data of your organization and easily categorize fact tables by business process.

To ensure that a fact-based model is built in a unified and standard manner, a business process is unique within a business unit and it exclusively belongs to a data domain. This standardizes naming and theme classification.

Dataphin allows you to view and manage the list of business processes created in a specific business unit or a specific project. You can also view and modify each business process.

#### Atomic metrics

An atomic metric is an abstraction of computing logic. To eliminate definition and development inconsistency, Dataphin introduces the concept of Design to Code. When a metric is defined, the statistical criteria (computing logic) is also defined. Re-engineering of the ETL process is not required, which increases development efficiency and ensures the consistency of statistical results.

Based on the complexity of computing logic, Dataphin categorizes atomic metrics into native atomic metrics and composite metrics.

- An example of a native atomic metric is payment amount.
- A composite metric is created based on the combination of atomic metrics. For example, the average sales per customer is calculated by dividing the total sales by the number of customers.

An atomic metric is unique within a business unit and has only one source logical table. The computing logic of an atomic metric is defined based on the fields of the source logical table model. This ensures that all statistical metrics are created in a unified and standard manner. The data domain of each logical table linked to the source logical table is retrieved to trace the data domains to which the atomic metric belongs. For example, an atomic metric may belong to multiple data domains. This ensures that names and logic are normalized and themes are classified in a standard manner.

## Business filters

**An atomic metric is the standardized definition of computing logic, and a business filter is the standardized definition of a query condition. Similar to an atomic metric, a business filter is unique within a business unit and has only one source logical table. The computing logic of a business filter is defined based on the fields of the source logical table model. This ensures that all statistical metrics are created in a unified and standard manner. The data domain of each logical table linked to the source logical table is retrieved to trace the data domains to which the business filter belongs. For example, a business filter may belong to multiple data domains. This ensures that names and logic are normalized and themes are classified in a standard manner.**

## Derived metrics

**Derived metrics are commonly used statistical metrics. To create derived metrics in a standard, regular, and clear manner, each derived metric is a calculation based on the following criteria:**

- **Atomic metric:** statistical criteria, that is, the computing logic.
- **Business filter:** the scope of business to be measured. It is used to filter the records that comply to specific business rules.
- **Statistical period:** a period during which statistics are collected, for example, the last 1 or 30 days.
- **Granularity:** a statistical object or perspective that defines the level of data aggregation. It can be considered as a grouping condition for aggregation, that is, GROUP BY clauses in SQL statements. Granularity is a combination of dimensions. For example, if a derived metric is a seller's turnover in a province, the granularity is the combination of the seller and the region dimensions.

**By combining the preceding parts, multiple derived metrics can be quickly created at a time while ensuring that the definitions and computing logic are clear without any duplication. This metric creation method is simple, available to all users, and does not require a high level of technical expertise. For example, business users can also complete metric creation. A derived metric is a concept that is based on the same level as a field. Each derived metric is unique and defined at the specified granularity level.**

## 32.4.5 Modeling

Dataphin provides systematic modeling and development functions to deeply implement the data warehouse theory. You can create business dimensions and business processes by using a top-down approach, and then enrich dimension tables, fact tables, aggregate tables, and the application data store layer. This process allows you to produce standardized data assets, which provides you with layered business data. The data standardization process can also optimize computation and storage.

### Logical dimension tables

A logical dimension table contains details about a dimension. Dataphin allows you to view and manage the list of created logical dimension tables, and to view and modify a specific logical dimension table.

### Logical fact tables

Dataphin supports using logical fact tables to model a specific business process (such as placing an order and paying for a commodity) or a state measure (such as account balance and inventory). A logical fact table is created in an optimized schema that is similar to a snowflake schema. Apart from measures and dimension-associated fields, this type of schema allows a fact table to also contain fact attributes. This reduces the complexity of the model design and makes it more user-friendly.

### Logical aggregate tables

The logical aggregate table model is an important data warehouse model. It contains two types of elements. The first type of element refers to various statistical values used to describe statistical granularity. The statistical values form a derived metric, for example, the sales in the last seven days. Granularity is a combination of several dimensions, such as the province and the product line dimensions. The second type of element refers to the attributes of the dimensions that constitute granularity. Examples of attributes are province name, product line name, product line level.

### Coding automation

After a logical dimension table, logical fact table, or logical aggregate table is published, Dataphin automatically designs the corresponding physical model, generates code and tasks to produce required data. Multiple tasks are usually

generated to convert a logical table to a physical model. If you want to view the task running logic, go to the Scheduling page.

### 32.4.6 Coding

Coding is an important data development method. This method can be used to achieve the same goal as building data models on graphical user interfaces. Dataphin allows you to edit scripts by using the coding method supported by your computing engine. You can submit the scripts to the scheduling system, which schedules the code tasks to produce data. You can also view historical versions of each code task. Multiple types of scripts are supported, such as SQL, Shell, and MapReduce scripts. The requirements for coding and configuration vary by script type. The requirements include syntax requirements and requirements for scheduling configuration. After a script is submitted and published, Dataphin creates a code task to run and produce data. In a directed acyclic graph (DAG), a task is also called a node. Dataphin supports the following operations for code task management: create, view, modify, and delete code tasks, edit scripts, configure task scheduling policies, publish tasks, and manage task versions.

#### Code editor

The code editor provides an online code editing interface to complete data development tasks. It supports SQL, MapReduce, Spark, and Shell programming.

#### Task scheduling configuration and publishing

- **Scheduling configuration**

You can configure the scheduling policy for one-time and recurring tasks. Tasks with a scheduling policy configured can be published. The system can check the integrity of task scheduling configurations. Only tasks with a complete scheduling configuration can be published. All published tasks are recurring tasks. You can choose Scheduling > Recurring Tasks and view the published recurring tasks in the left-side navigation pane.

- **Publish**

Members of a project can publish tasks if they have required permissions. Only a scheduling configuration with complete parameter settings, valid dependencies, and no circular dependencies can be published to create tasks. This guarantees that stable and orderly data production can be completed on schedule.

## Code management

**Dataphin supports various code operations to facilitate code file management and use. You can create, delete, update, rename, and view code files, and place code files in specific folders to categorize the code files.**

- **Manage files**

**Dataphin allows you to edit, delete, unpublish, and rename each code file. You can also view the publishing status, creator, and creation time of each code file. This facilitates easy creation, clear display, and systematic management of code files.**

- **Manage folders**

**When there are many code files, sort them in different folders to save and display these files in an orderly manner. You can create, rename, and delete folders, and move historical and new code files to specified folders for better management. Dataphin also supports hierarchical folder structures.**

## Collaborative programming

- **Manage node versions**

**Dataphin allows you to view historical task node versions. You can view the version number, submitter, submission time, and description. You can also view the code of each version to identify differences in code. Dataphin supports multiple node types, including MaxCompute\_SQL, MaxCompute MR, and Shell.**

- **Collaborative development**

**To achieve more efficient development by allowing collaboration between multiple developers, Dataphin provides a script locking mechanism, which prevents conflicts during collaborative development. This mechanism ensures that a line of code can only be edited by one user at a time. A user can steal the lock of another user to obtain the script editing permission. The user whose lock is stolen can obtain editing permission again by stealing the lock.**

### 32.4.7 Resource and function management

**Resource and function management assists code development. Data developers can upload local resources and configure task nodes for calling these resources to meet specific data processing requirements. These developers can also complete common data processing by using the built-in functions in the programming**

language supported by the computing engine. If a data logic (such as data conversion in compliance with a business logic) requires frequent processing and this cannot be achieved with the built-in functions, developers can define custom functions based on self-uploaded resources.

#### Resource management

Dataphin allows the data developers of a project to add, edit, and perform other operations on resources in the project. You can name and upload resource files, and then copy the resource file name to reference the resource file in the code. You can also delete unnecessary resource files.

- **Create and upload resource files**

By default, the following types of local resource files can be uploaded: XLS, DOC, TXT, CSV, JAR, Python, and other types (such as ZIP packages). New file types that are different from these types can be quickly added in three days by using the standard interface. Each resource file name is unique within a project. The file name and resource package cannot be changed after a resource file is submitted. Only one resource file can be uploaded each time, and the type of the uploaded file must be the same as the selected file type.

- **Reference resources**

You can copy and paste a resource file name to a specific position in the code editor, and write a statement to call this resource.

- **Update resources**

You can update the description of managed resources and delete existing resources to save storage space.

#### Function management

You can search, use, and manage functions. Functions are classified into two types: built-in functions of the system and user defined functions based on uploaded resources such as JAR and Python packages. You can extend user defined functions by referencing standard functions.

- **Create user defined functions**

Each user defined function must have a unique name within its project and cannot be renamed after being registered.

- **Reference functions**

**You can click Copy to copy the name of a built-in function or a user defined function, and then paste the name to a specific position in the code editor. Then, write a statement in the format of the sample command to process data.**

- **Update functions**

**You can update user defined functions by editing related information (except name) and delete unnecessary user defined functions.**

## **32.4.8 Scheduling and management**

**As a feature for routine maintenance and control in the late stage of data research and development, the scheduling center provides the list of all data processing tasks (such as recurring and one-time tasks), directed acyclic graphs (DAGs) of task dependency, list of task instances (such as recurring task instances, one-time task instances, and retroactive data generation task instances), dependencies of task instances, and state DAGs. With task scheduling and management, you can set the task execution sequence, split processes, achieve optimal distribution of machine resources, and discover abnormal tasks, ensuring that all the tasks can be stably and reliably executed on schedule. It also reports exceptions during task execution to ensure that exceptions can be handled in time. Currently, the scheduling and management module allows you to view tasks and manage task instances.**

### **Task list**

**Task list provides recurring and one-time tasks and DAG of task dependencies in different projects.**

### **Recurring tasks**

**For recurring tasks, you can view the task list, search for specific tasks, and view the dependencies of individual tasks. You can switch between different projects to view and search for tasks of specific projects, or perform a simple match by task node name or node ID. Dataphin supports secondary filtering of recurring tasks by task nodes of a specified user and task nodes published today, helping to narrow down the scope of tasks or locate tasks accurately for task scheduling and management.**



## One-time tasks

**For one-time tasks, you can view task lists, search for specific tasks, and view details of individual tasks. You can switch between different projects to view and search for tasks of specific projects, or perform a simple match by task node name or node ID. Dataphin supports secondary filtering of recurring tasks by task nodes of a specified user and task nodes published today, helping to narrow down the scope of tasks or locate tasks accurately for task scheduling and management.**

## Task instance management

**The task instance management provides recurring task instances, one-time task instances, and retroactive data generation instances in different projects, as well as details about task instances execution.**

## Recurring task instances

**For recurring task instances, you can view instance lists, search for specific instances, and view details of individual instances. You can view the running states of all common task instances and information for individual tasks, including their unique node IDs, node names, owners, task start times, end times, and durations . In addition, you can switch between different projects to view and search for task instances of specific projects, or perform a simple match by task node name or node ID. Dataphin supports secondary filtering of recurring task instances by my instances, instances with errors, unfinished nodes and task execution time, helping to narrow down the scope of instances or locate instances accurately for task instance maintenance.**

## One-time instances

**For one-time task instances, you can view instance lists, search for specific instances, and view details of individual instances. You can view the running states of all one-time task instances and information for individual tasks, including their unique node IDs, node names, owners, task start times, end times, and durations . In addition, you can switch between different projects to view and search for task instances of specific projects, or perform a simple match by task node name or node ID. Dataphin supports secondary filtering of one-time task instances by my instances and instances published today, helping to narrow down the scope of instances or locate instances accurately for task instance maintenance.**

## Retroactive data generation task instances

**In the list of retroactive data generation task instances, you can view task node names of retroactive data generation, time zones and states of retroactive data, and information about task nodes with retroactive data. The information includes the task node IDs, names, and owners, and retroactive duration. Searching and filtering of retroactive data generation task instances help you find a specific instance quickly and easily.**

## Logical tables

**This section allows you to search and view logical tables and the physical nodes contained in a logical table, as well as view the details of individual logical tables. You can switch between logical table tasks and logical table task instances to view details. By default, the DAG on the right pane of the logical table task shows all nodes contained in the logical table and the dependencies between internal nodes (including the "non-direct dependencies"). By default, the DAG on the right pane of the logical table task instance shows all node instances contained in the logical table and their statuses, including running, success, and failure statuses.**

## 32.4.9 Metadata center

**Dataphin provides powerful metadata management capabilities. It can collect and extract metadata from MaxCompute, Hadoop, Hive, MySQL, PostgreSQL, and Oracle data sources. It supports real-time tracing of metadata in the preceding computing and storage engines, and builds a unified metadata model by abstracting metadata from different types of storage engines. Dataphin supports the rapid expansion of multiple types of metadata and provides diverse metadata complying with unified standards. This ensures powerful and stable metadata for data maps and data governance.**

**The metadata center is the core foundation of data asset management. You must determine the following items when developing the metadata center:**

- **Metadata collection standard:** A unified data collection standard must be used to ensure the consistency among data information about model construction, data table creation, and data lineage. This improves the availability of metadata in retrieval and services.

- **Metadata freshness and quality:** The metadata output time and quality must be guaranteed to improve the freshness of asset management and application data, and the accuracy of data retrieval performed by developers.
- **Metadata model system:** A central public metadata model is built to ensure compatibility with various types of data and deliver a comprehensive data map service.

### 32.4.10 Data asset management

After data acquisition, integration, processing, and other processes, you can systematically manage the data with the data asset module. Based on OneData and data assets methodology, data asset management helps you to design application principle and leverage core technologies, including metadata acquisition, extraction, and processing technologies. This will classify and manage data in the form of assets, monitor data quality, and optimize resources, which can minimize the cost of data, maximize the value of data, and allow you to apply the value to your business.

Data asset management is implemented with a series of core technologies. The real-time event and subscription services enable real-time updates of tables, tasks, and other metadata. The rule engine ensures efficient and accurate judgment of data governance rules and the creation of health scoring models. Dynamic log analysis analyzes numerous production task execution logs and machine operations logs every day. Graph computing supports the analysis and establishment of data lineages. Onelog tracing analysis interlinks end-to-end metadata during data production, service, and consumption. The plug-in metadata access and processing architecture ensures compatibility between multiple computing and storage engines. Data asset management is a methodology of the Alibaba Group that includes a set of procedures, such as data collection, analysis, governance, application, and operation. It was developed based on the company's extensive experience with massive data management and covers the entire data lifecycle, including data creation, management, application, and destruction.

Data asset management involves two keywords: global and fusion. Global analysis is a process of checking all data and establishing a data asset map based on factors in the OneData system, including the dimensions, business processes, and correlations. This process describes data assets using a modeling language. Fusion is a process of analyzing the cost and value of data assets during production. This

process describes the functions of different data sets in the asset map based on the connectivity and contribution models.

Based on the data asset categories established after analysis of enterprise data assets, Dataphin's data map module integrates and analyzes user behavior data by using metadata profiling technology and a search engine. This enables efficient retrieval of an enterprise's data assets.

#### Asset overview

Enterprise data asset based on OneData can be displayed structurally in a chart. Components in different shapes in the chart represent business entities, whereas lines of different styles represent business relations between entities. This chart shows a clear overview of data structure in a single business unit.

#### Asset map

An asset map summarizes the relationships between dimensions and business processes in a data domain of a business unit to show the composition of your enterprise data, corresponding to the asset overview of the enterprise. In addition, the asset map provides an entry to efficient, fast, and accurate data search and exploration based on your self-initiated behaviors, such as searches, visits, and favorites.

### 32.4.11 Security management

The wide use of big data services makes data security an important issue. In China, the Cyber Security Law of the People's Republic of China was implemented on June 1, 2017. The Cyber Security Law encourages the development of network data security precautions and utilization technologies. EU General Data Protection Regulation (GDPR) was enacted on May 25, 2018. It aims to enhance the protection of data such as personal information. Dataphin focuses on intelligent development and management of data and places great importance on data security management. It provides comprehensive data security protection throughout the entire lifecycle (from data production to destruction). The protection is implemented by data access control, data isolation, and data security level classification. Other data protection methods include privacy compliance, data masking, and auditing of data security.

Data access control and data isolation require the highest priority in data security management. Dataphin provides management of data access permission requests

, approvals, and lifecycle. It supports data isolation for multi-tenancy and field level access control, and offers a data access authorization model based on access control lists (ACLs).

Dataphin establishes a comprehensive data security guarantee system covering the entire lifecycle of data. This system provides technologies and management measures to protect data from the perspectives of data access behaviors, data content, and data environment. During big data development and management, Dataphin works with the Alibaba Cloud data security management system to provide an available but invisible environment for secure big data exchange. Dataphin also supports field level access control, control of permission request approval processes, and tracing and auditing of data use behaviors. All these combined methods help to guarantee data security during the storage, transfer, and use of big data.

Dataphin offers a hierarchical permission control system and a full range of management, covering the request, approval, assignment, handover, and authentication of data access permissions.

#### Permission types

Dataphin provides data access control based on user roles and resources. This allows you to use Dataphin and access data in a secure and controllable manner.

- **Role privileges**

Dataphin provides account management mechanisms to obtain the super administrator and system members for centralized management of user operations. This controls the access methods of users at the platform level.

Dataphin also allows you to control resource access at the organizational level by using project management. This access control method is role-based access control. It assigns specific roles a set of data resource permissions. Users acquire permissions through the roles to which the users are assigned.

- **Resource permissions**

Dataphin provides a data access control mechanism to centrally manage user operations on project data resources. When each project is independently managed, and system members are isolated from resources, cross-project resource access can be controlled. This helps achieve data sharing by allowing users to use data of a specific project in another project without data migration.

## Permission management

- **Permission requests**

Data developers can find the required data table on the Data Map page and view the metadata details of this table. However, if they want to query data in the table, they must apply for permissions.

In a permission request process, Dataphin displays information about the requested data table by default, including the table type and the business unit to which the table belongs. Field metadata of the table is also displayed.

Dataphin supports permission requests that follow the principle of least privilege.

- Requests for field-level permissions are supported.
- Multiple options of permission validity period are provided. You can customize a date range or select 30 days, 90 days, 180 days, or 1 year as the validity period.
- You can describe the purposes for which you intend to use the requested permissions. The approver can determine whether to grant you the permissions based on the description.

- **Request management**

Dataphin allows you to view your requests and the status of the requests. You can click Details to view details of a request and click Cancel to cancel a request. After your request is approved, you can view your permission details, including the accessible fields.

- **Permission approval**

After a permission request is submitted, the system randomly assigns the ticket to an administrator of the project to which the requested data table belongs. The administrator needs to approve the request. Approvers can view details about the submitted requests on the My Approvals tab and decide whether to approve or reject the request.

- **Permission handover**

Users must hand over their permissions before shifting to another position or leaving the company. This ensures that related data and data production tasks can be handed over to appropriate staff. On the My Permissions page, you can

click Revoke to hand over your permissions to the project administrator. Then, Dataphin reclaims the permission.

### 32.4.12 Adhoc query

The adhoc query feature provides high-performance temporary data query and searching based on the Dataphin's powerful OneService engine. It supports both traditional simple query and theme-based query methods, and provides excellent code simplicity and fast query speeds.

#### Syntax

- Dataphin supports offline queries on all modeled logical tables. The intelligent engine selects the optimal physical table based on factors such as the output time and query performance.
- The join query function based on the snowflake schema makes SQL simpler and more intelligent.
- Dataphin supports queries on physical tables, logical tables, and combinations of physical tables and logical tables.
- Dataphin supports the syntax of multiple computing engines, such as MaxCompute SQL and Hive SQL.
- Dataphin provides intelligent prompt, pre-compiling, and functions beautifying for SQL.
- Dataphin provides permission management and user authentication at field level of logical or physical table.

#### Query execution

You can enter any query statements in a query script. The script editor provides intelligent prompts based on the input content, locates the required data table or field quickly, and check syntax automatically.

## 33 Elasticsearch

---

### 33.1 What is Elasticsearch?

Elasticsearch is a distributed search and data analytics service based on Lucene. It provides a distributed multi-tenant search engine that supports full text queries. This engine is based on a RESTful Web interface. Elasticsearch is developed based on Java. It is released as an open source product that complies with the Apache license terms and conditions. Elasticsearch is a mainstream search engine for enterprises. Elasticsearch is designed to serve cloud computing for real-time search. It is stable, reliable, fast, and easy to install and use.

Apsara Stack Elasticsearch provides two open source versions: Elasticsearch V5.5.3 and Elasticsearch V6.3.2. Apsara Stack Elasticsearch is designed to serve users in data search, data analytics, and other scenarios. Based on open source Elasticsearch, Apsara Stack Elasticsearch also supports enterprise-class permission management.

The default plug-ins provided by Apsara Stack Elasticsearch include but are not limited to the following:

- **IK analyzer:** an open source and lightweight Chinese analysis kit based on Java. The IK analyzer plug-in is very popular in open source communities for Chinese tokenization.
- **Smart Chinese analysis plug-in:** the default Lucene Chinese tokenizer.
- **ICU analysis plug-in:** a Lucene ICU tokenizer. ICU is a set of stable, tested, powerful, and easy to use libraries, providing Unicode and globalization support for applications.
- **Japanese (Kuromoji) analysis plug-in:** a Japanese tokenizer.
- **Stempel (Polish) analysis plug-in:** a French tokenizer.
- **Mapper attachments type plug-in:** an attachment-type plug-in which can parse files of different types into strings based on the Tika library.



## 33.2 Benefits

**Apsara Stack Elasticsearch provides the following benefits.**

- **Real-time data retrieval and analytics**

**Supports real-time retrieval and analysis of petabytes of data and responds in a few milliseconds.**

- **Stability and reliability**

**Alibaba Cloud Infrastructure as a Service (IaaS) supports disaster recovery and fault tolerance to guarantee the stability and reliability of data storage.**

- **Easy deployment and maintenance**

**Supports automated deployment with zero operations and maintenance costs, and integrates a system monitoring module.**

- **Visualized data analytics**

**Integrates the Kibana module for visualized data analytics and background management.**

- **Chinese tokenization**

**Provides built-in mainstream plug-ins, including the IK analyzer plug-in.**

- **Scalability**

**You can scale out an Elasticsearch cluster to hundreds of nodes, and upgrade or downgrade the hardware of these nodes as needed.**

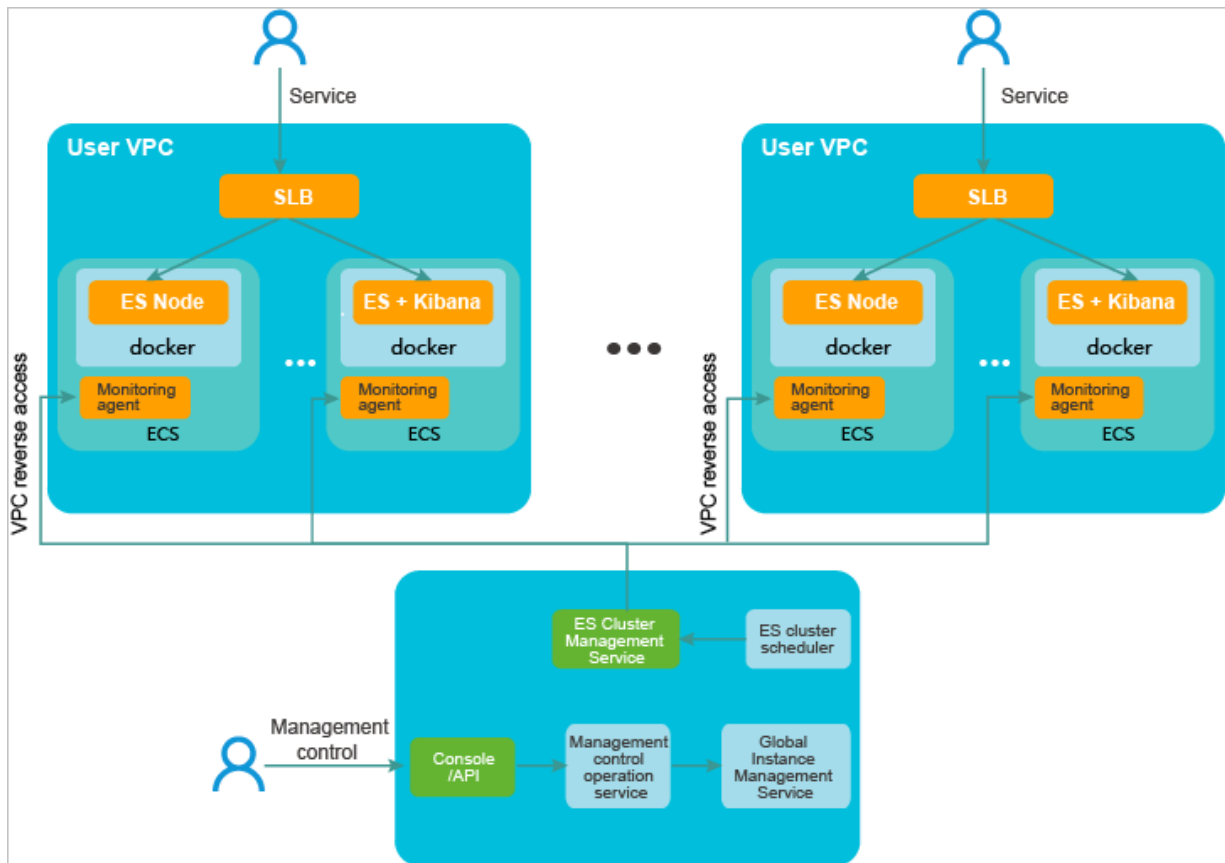
- **Technical support**

**Offers 24/7 technical support from the Alibaba Cloud technical support team, and provides documentation and training services.**

## 33.3 Architecture

**This topic describes the architecture of Apsara Stack Elasticsearch.**

**Taking the procedure of creating an Apsara Stack Elasticsearch instance as an example, the following figure shows the architecture.**



Submit the configuration of the Elasticsearch instance that you want to create from the Apsara Stack console or by calling the Elasticsearch API.

1. Select an Elastic Compute Service (ECS) instance type. The specified ECS instance is used as an Elasticsearch node and provides storage space.
2. The governance service retrieves the instance and storage space information from ECS, saves your request to the database, and then submits the request to the global instance management service.
3. The instance management service creates an Elasticsearch cluster configuration file based on the type of the request, and submits it to the Elasticsearch cluster management service.
4. The Apsara Stack Elasticsearch cluster management service is an offline processing system. Based on the request type, it runs a corresponding task state machine. The task state machine runs until the task reaches its final state.

For example, to create an instance, the Elasticsearch cluster management service labels the ECS instance, connects it to a VPC network, configures Server Load Balancer (SLB), and designates the cluster scheduler to manage the ECS instance.

The cluster scheduler then launches the Elasticsearch and Kibana processes on the ECS instance.

The Elasticsearch and Kibana processes run in containers on the ECS instance.

The monitor agent, an independent process, is in charge of collecting monitor metrics and then sending them to CloudMonitor through Log Service (SLS). Your instances are isolated by VPC networks. The governance service uses port mapping to establish reverse connections to your Elasticsearch instances to manage them.

## 33.4 Features

### 33.4.1 Kibana console

Apsara Stack Elasticsearch provides the Kibana console for you to scale your businesses. The Kibana console has been seamlessly integrated into Elasticsearch, allowing you to view the status of your Elasticsearch instances and manage these instances.

1. Log on to the Elasticsearch console.
2. Click the `instance ID` to go to the instance details page.
3. Click Kibana Console in the Basic Information area to log on to the Kibana console.

### 33.4.2 Restart an instance

The instance restart feature allows you to perform the restart or forced restart operation on your Elasticsearch cluster. Select an appropriate restart method based on your business scenario.

1. Log on to the Elasticsearch console.
2. Click an Elasticsearch instance ID or click Manage in the Actions column corresponding to an Elasticsearch instance to go to the Basic Information page.
3. Click Restart Instance at the upper-right corner. A dialog box is displayed.

#### 4. Select a restart method and click OK.

- **Restart:** The instance continues providing the highly efficient and highly available service (including at least one replica) during restart. This restart method takes a longer time compared with force restart.
- **Force Restart:** may cause the Elasticsearch cluster to provide unstable service during the process. However, this restart method is faster than the previous restart method.



##### **Note:**

Ensure that the health status of your Elasticsearch instance is green. The CPU utilization and memory usage of the Elasticsearch instance surge during the restart process. This may affect the stability of your service for a short period of time.



##### **Notice:**

When an Elasticsearch instance has a high disk usage, such as 85% or higher, the health status of the instance may change to yellow or red. In this case, the restart operation is disabled, and you can only perform forced restart.

- We recommend that you do not perform instance operations (such as node scaling, disk scaling, restart, password change, and configuration modification ) when the health status of your Elasticsearch instance is yellow or red. Perform these operations when the health status of your instance is green.
- If changing the configuration of an unhealthy instance that contains two or more nodes causes the instance to remain in the Initializing status, submit a ticket to resolve this issue.
- If performing the update, restart, scaling, or password reset operation on an Elasticsearch instance that contains only one node causes the service to become unavailable during the execution of the operation, create another Elasticsearch instance and migrate your service to the new instance.

### 33.4.3 Refresh

If part of information in the console (such as the status of a newly created Elasticsearch instance) is not refreshed in time, the console may fail to display

the information. In this case, you can manually refresh the Elasticsearch instance status on the page.

1. Log on to the Elasticsearch console.
2. Click an Elasticsearch instance ID or click Manage in the Actions column corresponding to an Elasticsearch instance to go to the Basic Information page.
3. Click Refresh at the upper-right corner to refresh the Elasticsearch instance status.

### 33.4.4 Basic information

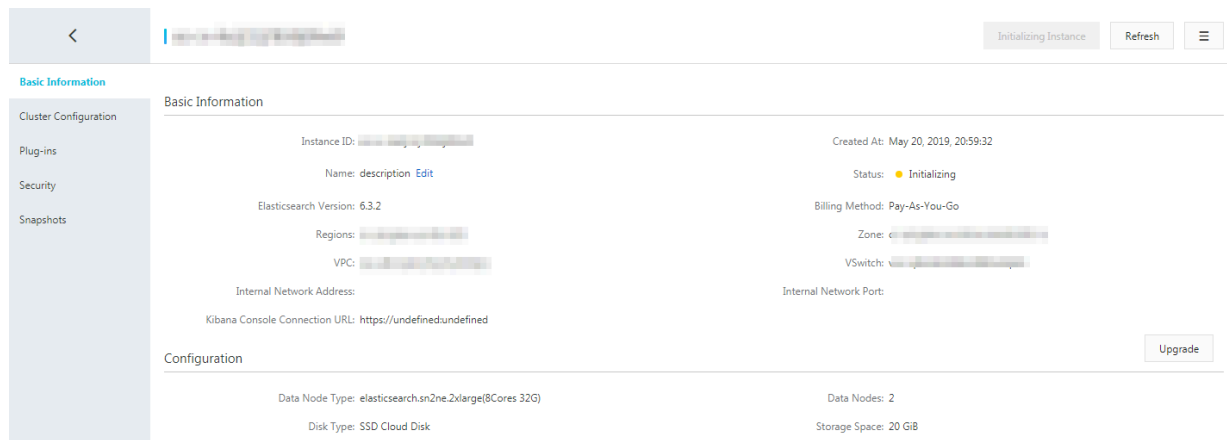


Table 33-1: Parameters

Parameter	Description
Upgrade	For more information, see <a href="#">Cluster upgrade</a> .
Name	The name of the instance. By default, the name of an Elasticsearch instance is the same as its ID. You can specify a name for an instance. You can also enter an instance name into the search box on the instances page to search for the instance.
Dedicated Master Node	Apsara Stack Elasticsearch dedicated master nodes. Dedicated master nodes are used to improve the stability of the instance. If you have purchased dedicated master nodes, this parameter displays Enabled on the basic information page.

Parameter	Description
Internal Network Address	You can use an internal network address to access an Elasticsearch instance from an ECS instance that is connected to the VPC network as the Elasticsearch instance. Supported ports include port 9200 for HTTP and port 9300 for TCP.
Kibana console	This parameter shows the address that is used to log on to the Kibana console.
Other parameters	For other parameters that are not described in this table, reference their parameter names.

### 33.4.5 Cluster upgrade

Apsara Stack Elasticsearch instances support upgrading the instance specification, storage space per data node, and number of nodes. Currently, you cannot downgrade Elasticsearch instances.

#### Procedure

1. Log on to the Elasticsearch console.
2. Click the ID of the target Elasticsearch instance or click Manage to navigate to the instance details page.
3. Click Upgrade on the right side of the page to open the cluster upgrade dialog box.
4. Edit the attributes of the Elasticsearch instance to meet your business demands, and then click OK.



#### Note:

- You can only edit one attribute at a time, such as the number of nodes, disk space per data node, or instance specification.
- If your business requires a cluster upgrade, we recommend that you make an upgrade assessment before upgrading the cluster.
- After you submit the upgrade order, the Elasticsearch instance will be billed based on the upgraded configuration.

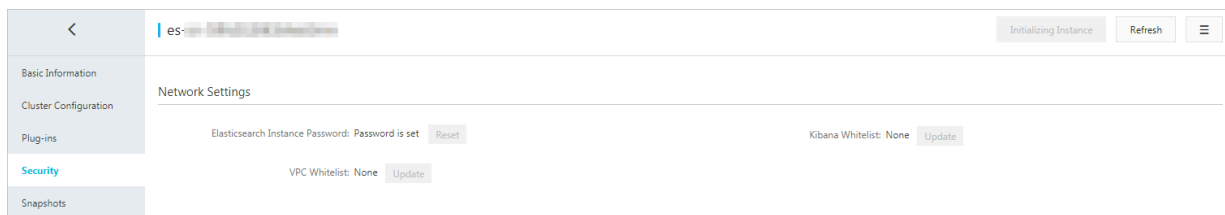
## 33.4.6 Elasticsearch cluster configurations

Elasticsearch cluster configurations include system configurations, language analysis configurations, and YML configurations.

### 33.4.6.1 Security

On the security page, you can reset the password of the Elasticsearch instance, edit the Kibana whitelist, and edit the VPC whitelist.

Figure 33-1: Security



Reset the Elasticsearch instance password

The reset Elasticsearch instance password feature only resets the password of the administrator account elastic. After you reset the password, you must use the new password to log on to the Elasticsearch instance and Kibana console.



#### Note:

- The password reset operation does not reset the password of other accounts that are used to log on to the Elasticsearch instance. We recommend that you do not use the elastic account to access your Elasticsearch instance.
- After you confirm the password reset operation, it takes up to five minutes for the new password to take effect.
- After you reset the password, Elasticsearch does not need to restart the instance to apply the new password.

Kibana whitelist

You can add IP addresses and CIDR blocks to the Kibana whitelist in the format of 192.168.0.1 and 192.168.0.0/24, respectively. Separate them with commas (,).

You can enter 127.0.0.1 to forbid all IPv4 addresses or enter 0.0.0.0/0 to allow all IPv4 addresses.

If your Elasticsearch instance is deployed in the China (Hangzhou) region, then you can add IPv6 addresses and CIRD blocks to the whitelist in the format of 2401:b180

`::1000:24::5` and `2401:b180:1000::/48`, respectively. Enter `::1` to forbid all IPv6 addresses or enter `::/0` to allow all IPv6 addresses.



**Note:**

The Kibana console can only be accessed from an ECS instance connected to the same VPC network as the Elasticsearch instance.

## VPC whitelist

You can add IP addresses and CIDR blocks to the VPC whitelist in the format of `192.168.0.1` and `192.168.0.0/24`, respectively. Separate them with commas (,). You can enter `127.0.0.1` to forbid all IPv4 addresses or enter `0.0.0.0/0` to allow all IPv4 addresses.



**Note:**

- By default, the VPC whitelist allows all IPv4 addresses.
- The VPC whitelist is used to control access from internal network addresses in VPC networks.

## 33.4.6.2 Word splitting

The synonym settings in the word splitting configuration is mainly applied to the Elasticsearch synonym dictionary. After you configure a synonym filter, new indexes are tokenized according to the latest synonym dictionary.

## Description

When you configure the synonym settings, you can define a synonym in each row of the UTF-8 encoded `.txt` file.



**Note:**

- After you upload and submit a synonym dictionary file, Elasticsearch does not need to restart the instance to update the dictionary. However, it takes a period of time for the new configuration to take effect.
- If you want to use the synonym dictionary to tokenize indexes that are created before the uploaded synonym dictionary file takes effect, then you must recreate the indexes and configure the synonym settings.



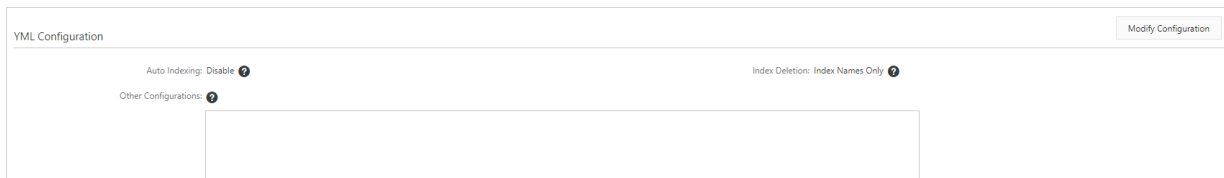
## Procedure

1. Upload and save a synonym dictionary file in the Apsara Stack Elasticsearch console. Make sure that the uploaded file takes effect.
2. When you create an index and configure the settings, you need to specify the "synonyms\_path": "analysis/your\_dict\_name.txt" path. Add a mapping for this index to configure synonyms for the specified field.
3. Verify the synonyms and upload a file for testing.

For more information, see [Configure synonyms](#).

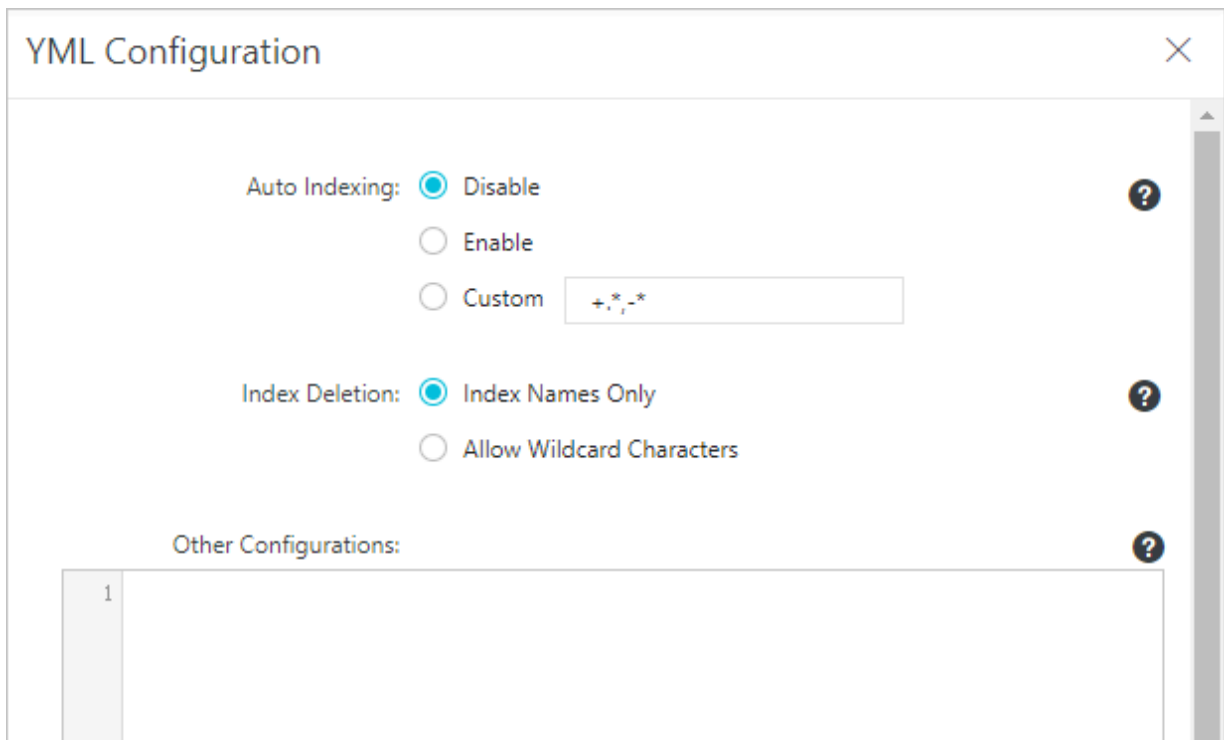
### 33.4.6.3 YML configuration

The YML configuration section in the Elasticsearch console displays the configuration of the current Elasticsearch instance.



## Modify configuration

Click **Modify Configuration** to modify the YML configuration.



- **Auto Indexing:** The auto-indexing feature allows the Elasticsearch instance to automatically create new indexes for documents uploaded to an instance if these documents do not have indexes. We recommend that you disable auto-indexing. Indexes created by this feature may not meet your requirements.
- **Index Deletion:** This feature specifies whether you need to specify the name of an index before you delete the index. If you select Allow Wildcard Characters, you can use wildcard characters to specify multiple indexes. After an index is deleted, it cannot be recovered. Proceed with caution.
- **Other Configurations:**

Some of the supported configuration items are as follows. For more information, see [Configuration parameters](#).

- `http.cors.enabled`
- `http.cors.allow-origin`
- `http.cors.max-age`
- `http.cors.allow-methods`
- `http.cors.allow-headers`
- `http.cors.allow-credentials`
- `reindex.remote.whitelist`
- `action.auto_create_index`
- `action.destructive_requires_name`

Reindex from a remote Elasticsearch instance. You can recreate indexes from a remote Elasticsearch instance that uses any Elasticsearch version. This feature allows you to migrate indexes from an earlier Elasticsearch version to the newly released Elasticsearch version. For more information about how to reindex from a remote Elasticsearch instance, see [Custom remote reindexing \(whitelisting\)](#).

### 33.4.6.3.1 Configuration parameters

The following table lists HTTP-based custom configuration parameters available in Elasticsearch.



#### Note:

These configuration parameters support only the static configuration mode, but not the hot deployment mode. To activate any of these configuration parameters, write it to the `elasticsearch.yml` configuration file.

Table 33-2: Configuration parameter description

Parameter	Description
<b>http.cors.enabled</b>	<p>The cross-origin resource sharing (CORS) configuration parameter. It is used to enable or disable CORS.</p> <ul style="list-style-type: none"> <li>• Elasticsearch can receive requests from browsers of resources in other domains. When this configuration parameter is set to <code>true</code>, Elasticsearch can process the <code>OPTIONS CORS</code> request.</li> <li>• If the domain information in the requests has been declared in <code>http.cors.allow-origin</code>, Elasticsearch adds <code>Access-Control-Allow-Origin</code> in the header to respond to the CORS request.</li> <li>• When this configuration parameter is set to <code>false</code> (the default value is <code>false</code>), Elasticsearch neglects the domain information in the header and does not add the <code>Access-Control-Allow-Origin</code> to the header to disable CORS.</li> <li>• If the client cannot send preflight requests that use the domain information header or does not check <code>Access-Control-Allow-Origin</code> in the header of the response from the server, secure CORS is affected.</li> <li>• If CORS is disabled for Elasticsearch, the client can try to send an <code>OPTIONS</code> request to check whether this response exists.</li> </ul>

Parameter	Description
<b>http.cors.allow-origin</b>	<p>The CORS resource configuration parameter. It can be used to configure to receive requests from which domains. No domain is allowed and the parameter is left blank by default.</p> <ul style="list-style-type: none"> <li>• If <code>/</code> is added before and after the parameter value, the configuration is identified as a regular expression.</li> <li>• You can use regular expressions to match HTTP- and HTTPS-based domain requests. For example, <code>/https?:\/\/localhost(:[0-9]+)?/</code> allows Elasticsearch to respond to the request satisfying this regular expression.</li> <li>• <code>*</code> is deemed as a valid configuration and indicates that the cluster supports CORS requests from any domain. This poses security risks to the Elasticsearch cluster.</li> </ul>
<b>http.cors.max-age</b>	<p>The browser can send an OPTIONS request to obtain the CORS configuration. <code>max-age</code> specifies how long the browser can retain the output result. The default value is 1,728,000 seconds (20 days).</p>
<b>http.cors.allow-methods</b>	<p>The request method configuration parameter. Valid values are <code>OPTIONS</code>, <code>HEAD</code>, <code>GET</code>, <code>POST</code>, <code>PUT</code>, and <code>DELETE</code>.</p>
<b>http.cors.allow-headers</b>	<p>The request header configuration parameter. Valid values are <code>X-Requested-With</code>, <code>Content-Type</code>, and <code>Content-Length</code>.</p>
<b>http.cors.allow-credentials</b>	<p>The credential configuration parameter. It is used to configure whether to return <code>Access-Control-Allow-Credentials</code> in the response header. If it is set to true, <code>Access-Control-Allow-Credentials</code> is returned. The default value is false.</p>

Parameter	Description
<b>reindex.remote.whitelist</b>	The remote host address whitelist that can access the cluster. Host-port pairs are allowed. Separate multiple pairs with commas (,) (such as otherhost:9200, another:9200, 127.0.10. *:9200, localhost:*). The whitelist only uses the host and port information for security policy configuration.
<b>action.auto_create_index</b>	The auto create index configuration parameter. When it is set to false, the auto create index feature is disabled.
<b>action.destructive_requires_name</b>	Indicates whether you need to specify the name of an index when you delete the index. When it is set to false (the default value), you can use regular expressions or <code>_all</code> to delete indexes. When it is set to true, you must specify index names to delete indexes, but cannot use <code>_all</code> or wildcards.

### 33.4.6.3.2 Custom remote reindexing (whitelisting)

The reindexing component allows you to reindex data from the remote Elasticsearch cluster. This feature is applicable to remote Elasticsearch instances of any version. It allows you to use the latest version to reindex data of old versions.

```
POST _REINDEX
{
  "SOURCE": {
    "REMOTE": {
      "HOST": "HTTP://OTHERHOST:9200",
      "USERNAME": "USER",
      "PASSWORD": "PASS"
    },
    "INDEX": "SOURCE",
    "QUERY": {
      "MATCH": {
        "TEST": "DATA"
      }
    }
  },
  "DEST": {
    "INDEX": "DEST"
  }
}
```

- host **must** contain the protocol, domain name, and port (such as `https://otherhost:9200`).

- **username and password are optional. If the remote Elasticsearch cluster needs to use the basic authorization scheme, the username-password pair is required. If you use the basic authorization scheme, we recommend that you use the HTTPS protocol. Otherwise, the password is transmitted as a text.**
- **The API can be called remotely only after the remote host address is declared in the `elasticsearch.yaml` configuration file by using the `reindex.remote.whitelist` attribute. `reindex.remote.whitelist` can use host-port pairs. Separate multiple pairs with commas (,) (such as, `otherhost:9200`, `another:9200`, `127.0.10.*:9200`, `localhost:*`). The whitelist does not identify the protocol and only uses the host and port information to configure security policies.**
- **If the host address is already listed in the whitelist, the query request is not verified or modified, but is directly sent to the remote Elasticsearch cluster.**

**Note:**

**Remote reindexing does not support manual or automatic slicing.**

The remote Elasticsearch cluster uses a stack to cache indexed data. The default maximum size is 100 MB. If a large document is involved in remote reindexing, set the size of the batch settings to a small value.

In the following example, the size of the batch settings is 10, which is the minimum value.

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "http://otherhost:9200"
    },
    "index": "source",
    "size": 10,
    "query": {
      "match": {
        "test": "data"
      }
    }
  },
  "dest": {
    "index": "dest"
  }
}
```

- **socket\_timeout: the timeout period for socket reading. The default value is 30 seconds.**
- **connect\_timeout: the connection timeout period. The default value is 1 second.**

**In the following example, `socket_timeout` is 1 minute and `connect_timeout` is 10 seconds.**

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "http://otherhost:9200",
      "socket_timeout": "1m",
      "connect_timeout": "10s"
    },
    "index": "source",
    "query": {
      "match": {
        "test": "data"
      }
    }
  },
  "dest": {
    "index": "dest"
  }
}
```

## 34 DataHub

---

### 34.1 What is DataHub?

#### 34.1.1 Overview

**DataHub is a real-time data distribution platform designed to process streaming data.**

**You can publish and subscribe to applications for 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 Alibaba Cloud 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 also supports synchronizing streaming data to several Alibaba Cloud services such as MaxCompute and OSS.**

**The features of DataHub are described as follows:**

- **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.
- **Checkpoint-based data restoration:** DataHub supports saving checkpoints in the system. You can restore data from any saved checkpoint when your application fails.
- **Data synchronization:** Data in DataHub can be automatically synchronized to other Alibaba Cloud platforms, including MaxCompute, Object Storage Service (OSS), AnalyticDB, ApsaraDB RDS for MySQL, Table Store, and Elasticsearch.



- **Scalable topics:** DataHub allows you to scale in or out a topic by splitting a shard into two or merging two shards.

### 34.1.2 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.**

### 34.1.3 Highlights

The highlights of DataHub features are described as follows:

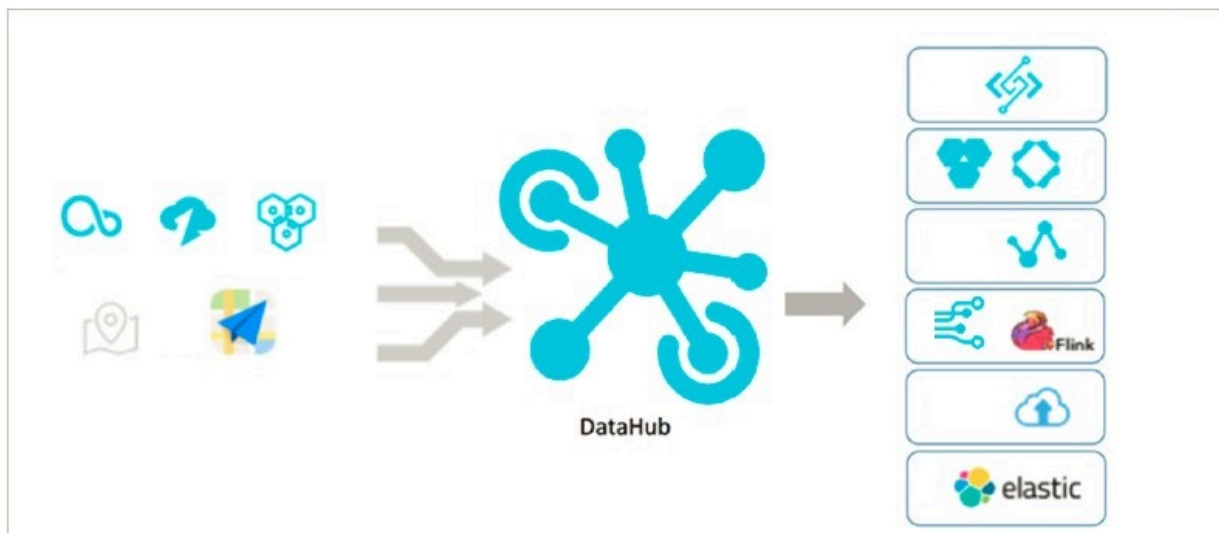
Table 34-1: Highlights

Highlight	Description
Data security	DataHub ensures data security based on the Alibaba Cloud RAM system.
Simple O&M	DataHub automatically deactivates and recovers problematic nodes before reactivating the nodes.
Resource isolation	DataHub isolates resources between tenants.
Connection with various Alibaba Cloud services	DataHub can be used with a variety of other Alibaba Cloud services.
Scalability	The processing capacity of DataHub is automatically expanded without affecting your services. The scalability is verified during the service peak of Double 11.
Read/write performance	Records written into DataHub can be consumed repeatedly within the time-to-live of the records .
High availability	DataHub offers various high availability solutions.
Seamless integration with Alibaba Cloud services	DataHub is seamlessly integrated with various Alibaba Cloud services.

### 34.1.4 Scenarios

#### Data uploading

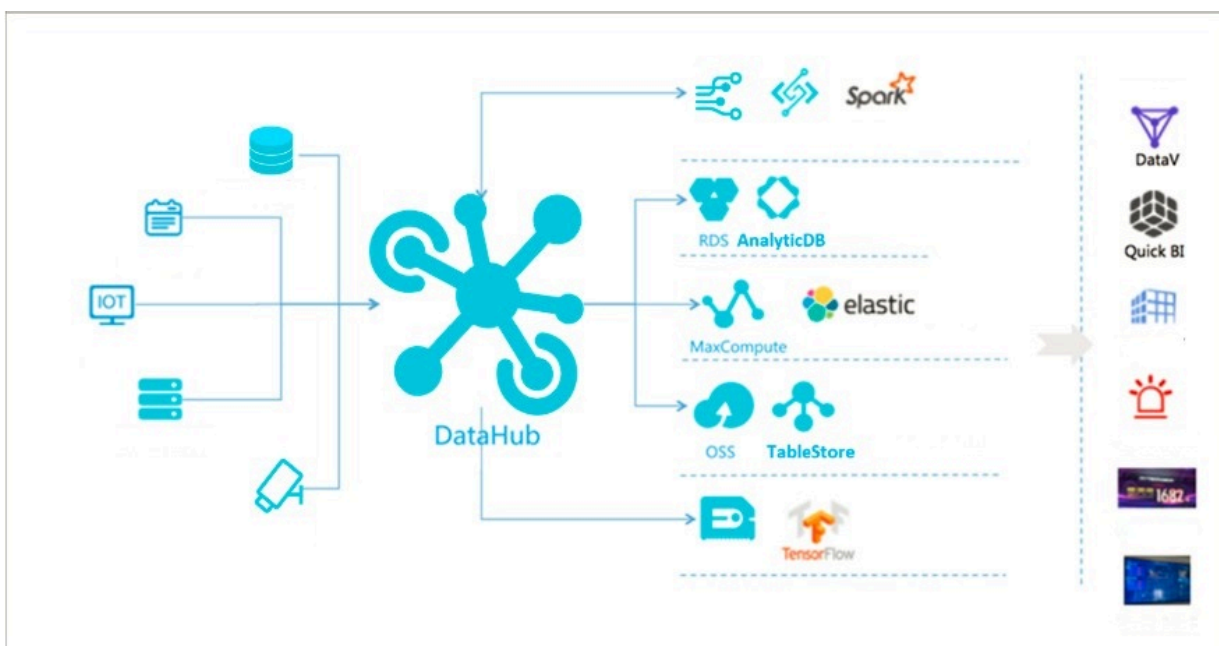
Figure 34-1: Data uploading



**DataHub is connected to other Alibaba Cloud services, saving you the trouble of uploading the same data to different platforms.**

#### Data collection

Figure 34-2: 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**

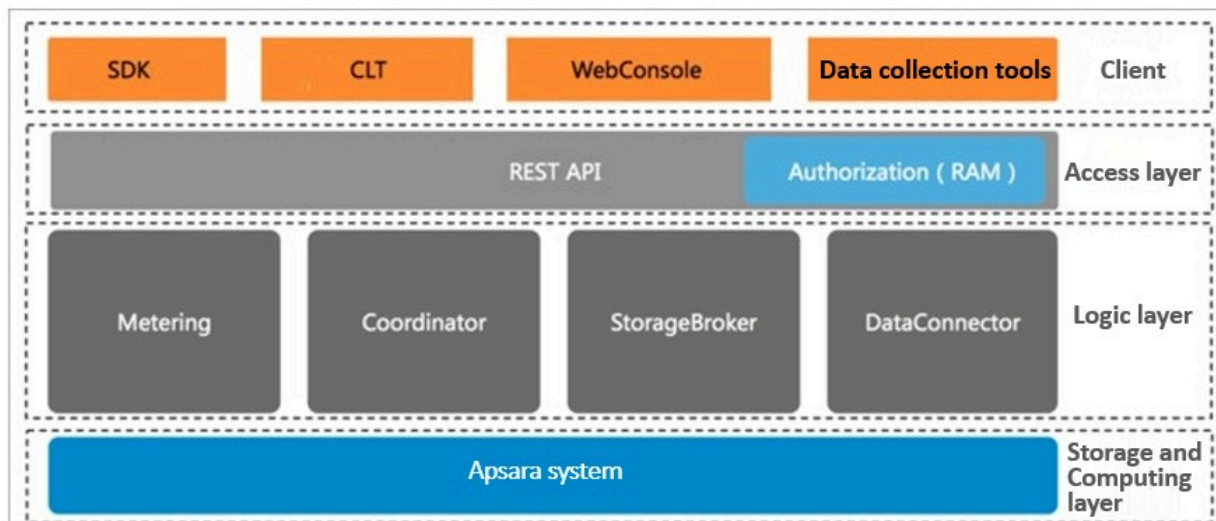
binary log collection from Data Transmission Service (DTS) and Oracle GoldenGate (OGG). DataHub also supports the collection of surveillance videos through GB28181.

## 34.2 Architecture

### 34.2.1 Feature oriented architecture

*Figure 34-3: Feature oriented architecture of DataHub* shows the feature oriented architecture of DataHub.

Figure 34-3: Feature oriented architecture of DataHub



The architecture of DataHub consists of four layers: client, access layer, logical layer, and storage and scheduling layer.

#### Client

DataHub supports the following types of clients:

- **SDKs:** DataHub provides a variety of SDKs for C++, Java, Python, Ruby, and Go.
- **Command line tool (CLT):** 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 shard details, monitor topic performance, and manage DataConnector.
- **Data collection tools:** Logstash, Fluentd, and Oracle GoldenGate (OGG).

### Access layer

**DataHub can be accessed through HTTP and HTTPS. DataHub supports RAM authorization and horizontal scaling of topic performance.**

### Logical layer

**The logical layer handles the key features of DataHub, including project and topic management, data read and write, checkpoint-based data restoration, traffic statistics, and data archives. Based on these key features, the logical layer is composed of the following modules: StorageBroker, Metering, Coordinator, and DataConnector.**

- **StorageBroker:** Enables the reading and writing of data in DataHub. Adopts the log file storage model of the Apsara Distributed File System, halving the read/write volume compared with the transfer of write-ahead logs. Stores three copies of data to ensure that no data is lost if a server fault occurs. Supports disaster recovery between data centers. Supports data write caching to ensure efficient consumption of real-time data. Supports independent read caching of historical data to enable concurrent consumption of the same data.
- **Metering:** Supports shard-level billing based on the consumption period.
- **Coordinator:** Supports checkpoint-based data restoration. Provides 150,000 QPS per node. Supports horizontal scaling of the processing capacity.
- **DataConnector:** Supports automatic data synchronization from DataHub to other Alibaba Cloud services, including MaxCompute, Object Storage Service (OSS), AnalyticDB, ApsaraDB RDS for MySQL, Table Store, and Elasticsearch.

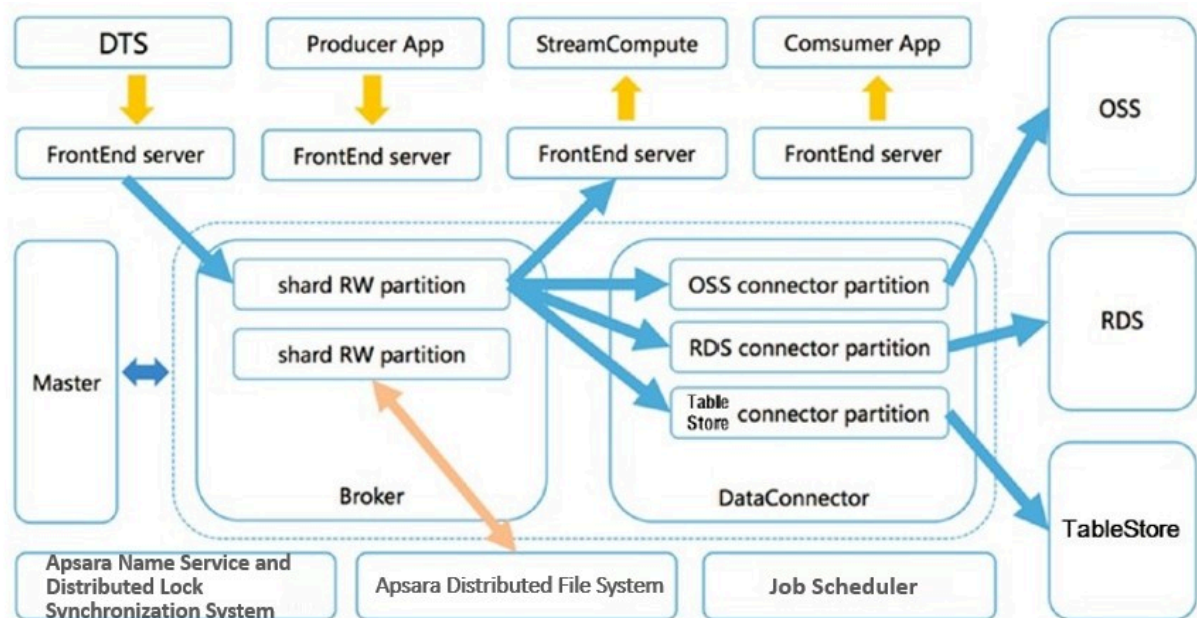
### Storage and scheduling layer

- **Storage:** Based on the log file storage model of the 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 recording time of the data.
- **Scheduling:** Based on the scheduling module of Job Scheduler, DataHub assigns shards to nodes based on the traffic that occurs 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.

## 34.2.2 Technical architecture

*Figure 34-4: Technical architecture of DataHub* shows the technical architecture of DataHub.

Figure 34-4: Technical architecture of DataHub



The figure shows the process from data ingestion to consumption.

1. A shard is the smallest unit of data management in DataHub, and is a first-in, first-out (FIFO) collection of records.
2. Data in each shard is stored in a set of log files on the Apsara Distributed File System.
3. The master distributes each shard to a broker. Each broker is responsible for the read and write operations of multiple shards.
4. The frontend server locates a broker based on the project, topic, and shard information specified in the request and forwards the request to the broker.
5. DataConnector reads data from the broker and forwards the data to other Alibaba Cloud services.

### Data collector

You can write data into DataHub from applications developed by using SDKs and from data collection tools such as LogStash, Fluentd, and Oracle GoldenGate.

You can also write data by using Data Transmission Service (DTS) and Realtime Compute.

#### Frontend server

**Frontend servers constitute the access layer and support horizontal scaling. You can call RESTful API operations to access DataHub. RAM authorization is supported**

.

#### Master

**The master handles metadata management and shard scheduling. It supports create, read, update, and delete operations on projects and topics. The master also supports split and merge operations on shards.**

#### Broker

**Brokers handle read and write operations on each shard including data indexing, caching, and file organization and management.**

#### DataConnector

**DataConnector forwards data in DataHub to other Alibaba Cloud services.**

**DataConnector provides different features for various destination services. These features include automatically creating partitions in MaxCompute and converting data streams into files stored in OSS.**

## 34.3 Features

### 34.3.1 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.**

### 34.3.2 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.**

### 34.3.3 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, including MaxCompute, OSS, Elasticsearch, ApsaraDB RDS for MySQL, AnalyticDB, and Table Store.

You can configure the DataConnector so that the data you write to DataHub can be used in other cloud platforms. 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.

Table 34-2: Destination platforms

Destination platform	Timeliness	Description
MaxCompute	Near real-time . Latency: 5 minutes.	The column names and data types in the source topic must be the same as those in MaxCompute. The MaxCompute table must have one or more corresponding partition columns.
OSS	Real-time	Records are synchronized to the specified bucket in OSS and are saved as CSV files.
Elasticsearch	Real-time	Records are synchronized to the specified index in Elasticsearch. Records may not be synchronized in the order of the recording time. If you want to synchronize data in the order of the recording time, you must write the records with the same partition key into the same shard.
ApsaraDB RDS for MySQL	Real-time	Records are synchronized to the specified table in ApsaraDB RDS for MySQL.
AnalyticDB	Real-time	Records are synchronized to the specified table in AnalyticDB.



Destination platform	Timeliness	Description
Table Store	Real-time	Records are synchronized to the specified table in Table Store.

### 34.3.4 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.